



Patent Application of

Jeff S. Eder

for

**AN AUTOMATED METHOD OF AND SYSTEM FOR IDENTIFYING, MEASURING AND
ENHANCING CATEGORIES OF VALUE FOR A VALUE CHAIN**

**CONTINUATION AND
CROSS REFERENCE TO RELATED PATENT**

This application is a continuation of application number 09/421,553, filed July 22, 1999. The subject matter of this application is also related to the subject matter of U.S. Patent 5,615,109 for "Method of and System for Generating Feasible, Profit Maximizing Requisition Sets", by Jeff S. Eder, the disclosure of which is incorporated herein by reference.

BACKGROUND OF THE INVENTION

This invention relates to a method of and system for business valuation, more particularly, to an automated system that identifies, evaluates and helps improve the management of the categories of value for a value chain and for each enterprise in the value chain on a continual basis.

The internet has had many profound effects on global commerce. The dramatic increase in the use of email, the explosion of e-commerce and the meteoric rise in the market value of internet firms like eBay, Amazon.com and Yahoo! are some of the more visible examples of the impact it has had on the American economy. Another impact of the internet has been that it has enabled the "virtual integration" of companies in different locations and different industries. Companies can now join together in a matter of days with essentially no investment to form a "virtual value chain" for delivering products and services to consumers.

The virtual value chain may appear to the consumer as a single entity, when in reality a number of enterprises from different continents have joined together to complete the preparation and delivery of the good or service that is ultimately being purchased. Virtual value chains allow each firm in the value chain to focus on their own specialty, be it manufacturing, design, distribution or marketing while reaping the benefits of the increased scale and scope inherent in the alliance. Enabled by the low cost communication capability provided by the internet, the virtual value chain is really just an extreme form of a phenomenon that has been sweeping American industry for many years – the electronic linkage of businesses.

Despite the widespread acceptance and use of "virtual value chains" as a mechanism for efficiently and effectively responding to customer demands, there is no known method or system for systematically evaluating the value of these new types of organizations. In a similar manner there is no known method or system for evaluating the contribution of the different enterprises in the "virtual value chain".

The need for a systematic approach for evaluating "virtual value chains" is just part of a larger need that has recently appeared for a new method for systematically evaluating the financial performance of a commercial business. The need for a new approach has been highlighted in the past two years by the multi-billion dollar valuations being placed on internet companies like Amazon.com, E trade and eBay that have never earned a dollar of profit and that have no prospect of earning a dollar of profit any time soon. The most popular traditional approaches to valuation are all based on some multiple of accounting earnings (a price to earnings ratio or P/E ratio) – with no corporate earnings in the past or the foreseeable future – these methods are of course useless in evaluating the new companies.

The inability of traditional methods to provide a framework for analyzing "virtual value chains" and internet firms are just two glaring examples of the weakness of traditional financial systems. Numerous academic studies have demonstrated that accounting earnings don't fully explain changes in company valuations and the movement of stock prices. Many feel that because of this traditional accounting systems are driving information-age managers to make the wrong decisions and the wrong investments. Accounting systems are "wrong" for one simple reason, they track tangible assets while ignoring intangible assets. Intangible assets such as the skills of the workers, intellectual property, business infrastructure, databases, and relationships with customers and suppliers are not measured with current accounting systems. This oversight is critical because in the present economy the success of an enterprise is determined more by its ability to use its intangible assets than by its ability to amass and control the physical ones that are tracked by traditional accounting systems.

Consultants from McKinsey & Company recently completed a three year study of companies in 10 industry segments in 12 countries that confirmed the importance of intangible assets as enablers of new business expansion and profitable growth. The results of the study, published in the book The Alchemy of Growth, revealed three common characteristics of the most successful businesses in today's economy:

1. They consistently utilize "soft" or intangible assets like brand names, customers and employees to support business expansion;

2. They systematically generate and harvest real options for growth; and
3. Their management focuses on 3 distinct "horizons"— short term (1 – 3 years), growth (3 – 5 years out) and options (beyond 5 years).

The experience of several of the most important companies in the U.S. economy, IBM, General Motors and DEC, in the late 1980's and early 1990's illustrates the problems that can arise when intangible asset information is omitted from corporate financial statements and companies focus only on the short term horizon. All three companies were showing large profits using current accounting systems while their businesses were deteriorating. If they had been forced to take write-offs when the declines in intangible assets were occurring, the problems would have been visible to the market and management would have been forced to act to correct the problems much more quickly than they actually did. These deficiencies of traditional accounting systems are particularly noticeable in high technology companies that are highly valued for their intangible assets and their options to enter growing markets rather than their tangible assets.

The appearance of a new class of software applications, soft asset management applications, is further evidence of the increasing importance of "soft" or intangible assets. Soft asset management applications (or systems) include: alliance management systems, brand management systems, customer relationship management systems, channel management systems, intellectual property management systems, process management systems and vendor management systems. While these systems enhance the day to day management of the individual "soft" assets, there is currently no mechanism for integrating the input from each of these different systems in to an overall organization or enterprise asset management system. As a result, the organization or enterprise can be (and often is) faced with conflicting recommendations as each system tries to optimize the asset it is focused on without considering the overall financial performance of the organization or enterprise.

A number of people have suggested using business valuations in place of traditional financial statements as the basis for measuring and managing financial performance. Unfortunately, using current methods, the valuation of a business is a complex and time-consuming undertaking. Business valuations determine the price that a hypothetical

buyer would pay for a business under a given set of circumstances. The volume of business valuations being performed each year is increasing significantly. A leading cause of this growth in volume is the increasing use of mergers and acquisitions as vehicles for corporate growth. Business valuations are frequently used in setting the price for a business that is being bought or sold. Another reason for the growth in the volume of business valuations has been their increasing use in areas other than supporting merger and acquisition transactions. For example, business valuations are now being used by financial institutions to determine the amount of credit that should be extended to a company, by courts in determining litigation settlement amounts and by investors in evaluating the performance of company management.

Income valuations are the most common type of valuation. They are based on the premise that the current value of a business is a function of the future value that an investor can expect to receive from purchasing all or part of the business. In these valuations the expected returns from investing in the business and the risks associated with receiving the expected returns are evaluated by the appraiser. The appraiser then determines the value whereby a hypothetical buyer would receive a sufficient return on the investment to compensate the buyer for the risk associated with receiving the expected returns. One difficulty with this method is determining the length of time the company is expected to generate the expected returns that drive the valuation. Most income valuations use an explicit forecast of returns for some period, usually 3 to 5 years, combined with a "residual". The residual is generally a flat or uniformly growing forecast of future returns that is discounted by some factor to estimate its value on the date of valuation. In some cases the residual is the largest part of the calculated value.

One of the problems inherent in a steady state "residual" forecast is that returns don't continue forever. Economists generally speak of a competitive advantage period or CAP (hereinafter referred to as CAP) during which a given firm is expected to generate positive returns. Under this theory, value is generated only during the CAP. After the CAP ends, value creation goes to zero or turns negative. Another change that has been produced by the internet economy is that the CAP for most businesses is generally thought to be shrinking with the exception of companies whose products possess network externalities that tie others to the company and its products or services. These latter companies are

thought to experience increasing returns as time goes by rather than having a finite CAP. Because the CAP is hard to calculate, it is generally ignored in income valuations however, the simplification of ignoring the CAP greatly reduces the utility of the valuations that are created with large residuals.

When performing a business valuation, the appraiser is generally free to select the valuation type and method (or some combination of the methods) in determining the business value. The usefulness of these valuations is limited because there is no correct answer, there is only the best possible informed guess for any given business valuation. The usefulness of business valuations to business owners and managers is restricted for another reason - valuations typically determine only the value of the business as a whole. To provide information that would be useful in improving the business, the valuation would have to furnish supporting detail that would highlight the value of different categories of value within the business. An operating manager would then be able to use a series of business valuations to identify categories within a business that have been decreasing in value. This information could also be used to help identify corrective action programs and to track the progress that these programs have made in increasing business value. This same information could also be used to identify categories that are contributing to an increase in business value. This information could be used to identify categories where increased levels of investment would have a significant favorable impact on the overall health of the business.

Even when intangible assets have been considered, the limitations in the existing methodology have severely restricted the utility of the valuations that have been produced. All known prior efforts to value intangible assets have been restricted to independent valuations of different types of intangible assets (similar to the individual soft asset management systems discussed previously). Intangible assets that have been valued separately in this manner include: brand names, customers and intellectual property. Problems associated with existing methods for valuing intangible assets include:

1. interactions between the different intangible assets are ignored,
2. the actual impact of the asset on the enterprise isn't measured,

3. the relative strength of the intangible asset within the industry is just as important (and in some cases more important) than any absolute measure of its strength, and
4. there is no systematic way for determining the life of the assets.

Typically, intangible asset valuations also ignore the real options for growth that are intimately inter-related and dependent upon the intangible assets being evaluated. In addition to having a direct influence on the valuation of a given real option the enterprise may possess, intangible assets can affect the market's perception of which company is likely to receive the lions share of future growth in a given industry. This, in turn affects the allocation of industry options to the market price for equity in the enterprise.

The lack of a consistent, well accepted, realistic method for measuring all the categories of business value also prevents some firms from receiving the financing they need to grow. Most banks and lending institutions focus on book value when evaluating the credit worthiness of a business seeking funds. As stated previously, the value of many high technology firms lies primarily in intangible assets and real options that aren't visible under traditional definitions of accounting book value. As a result, these businesses generally aren't eligible to receive capital from traditional lending sources, even though their financial prospects are generally far superior to those of companies with much higher tangible book values.

In light of the preceding discussion, it is clear that it would be advantageous to have an automated financial system that valued all the assets and options for a given organization. Ideally, this system would be capable of generating detailed valuations for businesses in new industries while prioritizing and coordinating the management of the different soft assets that the organization is tracking.

SUMMARY OF THE INVENTION

It is a general object of the present invention to provide a novel and useful system that continuously calculates and displays a comprehensive and accurate valuation for all the categories of value for a virtual organization that overcomes the limitations and drawbacks of the existing art that were described previously.

A preferable object to which the present invention is applied is the valuation and coordinated management of the different categories of value within an organization that consists of two or more commercial enterprises that have come together to form a "virtual value chain" for the purpose of delivering products or services to customers where a large portion of the organization's business value is associated with intangibles and real options.

The present invention also provides the ability to calculate and display a comprehensive and accurate valuation for the categories of value for each commercial enterprise within the virtual value chain. The ability to "drill down" for more detailed analysis extends to each element of value within each enterprise in the "virtual value chain" as illustrated in Table 1.

Table 1

| <u>Level</u> | <u>Valuation Categories</u> |
|---------------------|--|
| Organization | Current Operation: Assets/Liabilities Current Operation: Enterprise Contribution & Joint: Real options/Contingent Liabilities |
| Enterprise | Current Operation: Assets/Liabilities Current Operation: Elements of Value Real Options/Contingent Liabilities & Market Sentiment |
| Element of Value | Sub-elements of value |

The present invention eliminates a great deal of time-consuming and expensive effort by automating the extraction of data from the databases, tables, and files of existing computer-based corporate finance, operations, human resource and "soft" asset

management system databases as required to operate the system. In accordance with the invention, the automated extraction, aggregation and analysis of data from a variety of existing computer-based systems significantly increases the scale and scope of the analysis that can be completed. The system of the present invention further enhances the efficiency and effectiveness of the business valuation by automating the retrieval, storage and analysis of information useful for valuing categories of value from external databases and publications and the internet. Uncertainty over which method is being used for completing the valuation and the resulting inability to compare different valuations is eliminated by the present invention by consistently utilizing the same set of valuation methodologies for valuing the different categories of organization value as shown in Table 2.

Table 2

| Organization Categories of Value | | Valuation methodology |
|--|--|---|
| • Total current-operation value (COPTOT): | | Income Valuation |
| Current Operation Assets/Liabilities: | Cash & Marketable Securities (CASH), Inventory (IN), Accounts Receivable (AR), Prepaid Expenses (PE), Other Assets (OA); Accounts Payable (AP), Notes Payable (NP), Other Liabilities (OL) | GAAP for portion of assets/liabilities from each enterprise that are devoted to the organization |
| Current Operation Assets/Liabilities: | Production Equipment (PEQ), Other Physical Assets (OPA) | Replacement Value for portion of assets from each enterprise that are devoted to the organization |
| Current Operation Enterprise Contribution: | Enterprise contribution to virtual value chain (VVCC) | System calculated value |
| Current Operation Enterprise Contribution: | General going concern element of value (GGCV) | $GGCV = COPTOT - CASH - AR - IN - PE - PEQ - OPA - OA - VVCC$ |
| • Real options/Contingent Liabilities | | Real option algorithms + allocation of industry real options based on relative industry position |

* The user also has the option of specifying the total value

The present invention takes a similar approach to enterprise value analysis by consistently utilizing the same set of valuation methodologies for valuing the different categories of enterprise value as shown in Table 3.

Table 3

| Enterprise Categories of Value | | Valuation methodology |
|--|--|---|
| • Total current-operation value (COPTOT): | | Income Valuation |
| Current-operation Assets/Liabilities: | Cash & Marketable Securities (CASH), Inventory (IN), Accounts Receivable (AR), Prepaid Expenses (PE), Other Assets (OA), Accounts Payable (AP), Notes Payable (NP), Other Liabilities (OL) | GAAP |
| Current-operation Assets/Liabilities: | Production Equipment (PEQ), Other Physical Assets (OPA) | Replacement Value |
| Current Operation Elements of Value (EV): | Alliances, Brand Names, Channel Partners, Customers, Employees, Industry Factors*, Infrastructure, Intellectual Property, Information Technology, Processes and Vendors | System calculated value |
| Current Operation Element of Value: | General going concern (GCV) | $GCV = COPTOT - CASH - AR - IN - PE - PEQ - OPA - OA - \sum EV$ |
| • Real options/Contingent Liabilities | | Real option algorithms + allocation of industry real options based on relative strength of elements of value (EV) |
| • Market Sentiment | | Enterprise Market Value – (COPTOT + \sum Real option Values) |

* Note: Industry Factors (regulation, concentration, etc.) are analyzed like an element of value

There is no market sentiment calculation at the organization level because the market value of each enterprise in the organization generally includes non-value chain related activities and the firm level market sentiment for each enterprise can not readily be subdivided in to value chain and non-value chain sentiment.

The market value of each enterprise in the organization is calculated by adding the market value of all debt and equity as shown in Table 4.

Table 4

| |
|--|
| Enterprise Market Value = |
| Σ Market value of enterprise equity |
| + |
| Σ Market value of company debt |

One benefit of the novel system is that the market value of every enterprise in the organization is subdivided in to at least three distinct categories of value: current operation assets, elements of value and real options. As shown in the table 5, these three value categories match the three distinct "horizons" for management focus the McKinsey consultants reported on in The Alchemy of Growth.

Table 5

| System Value Categories | Three Horizons |
|--------------------------------|-----------------------|
| Current Operation Assets | Short Term |
| Elements of Value | Growth |
| Real Options | Options |

The utility of the valuations produced by the system of the present invention are further enhanced by explicitly calculating the lives of the different elements of value as required to remove the inaccuracy and distortion inherent in the use of a large residual.

As shown in Tables 2 and 3, growth opportunities and contingent liabilities are valued using real option algorithms. Because real option algorithms explicitly recognize whether or not an investment is reversible and/or if it can be delayed, the values calculated using these algorithms are more realistic than valuations created using more traditional approaches like Net Present Value. The use of real option analysis for valuing growth opportunities and contingent liabilities (hereinafter, real options) gives the present invention a distinct advantage over traditional approaches to business valuation.

The innovative system has the added benefit of providing a large amount of detailed information concerning both tangible and intangible elements of value. Because

intangible elements are by definition not tangible, they can not be measured directly. They must instead be measured by the impact they have on their surrounding environment. There are analogies in the physical world. For example, electricity is an "intangible" that is measured by the impact it has on the surrounding environment. Specifically, the strength of the magnetic field generated by the flow of electricity through a conductor is used to determine the amount of electricity that is being consumed. The system of the present invention measures intangible elements of value by identifying the attributes that, like the magnetic field, reflect the strength of the element in driving the components of value (revenue, expense and change in capital) and are easy to measure. Once the attributes related to each element's strength are identified, they are summarized into a single expression (a composite variable or vector). The vectors for all elements are then evaluated to determine their relative contribution to driving each of the components of value. The system of the present invention calculates the product of each element's relative contribution and forecast life to determine the contribution to each of the components of value. The contributions to each component of value are then added together to determine the value of each element (see Table 7).

The system also gives the user the ability to track the changes in categories of value by comparing the current valuations to previously calculated valuations. As such, the system provides the user with an alternative to general ledger accounting systems for tracking financial performance. To facilitate its use as a tool for improving the value of a commercial enterprise, the system of the present invention produces reports in formats that are similar to the reports provided by traditional accounting systems. The method for tracking the categories of value for a business enterprise provided by the present invention eliminates many of the limitations associated with current accounting systems that were described previously.

BRIEF DESCRIPTION OF DRAWINGS

These and other objects, features and advantages of the present invention will be more readily apparent from the following description of the preferred embodiment of the invention in which:

FIG. 1 is a block diagram showing the major processing steps of the present invention;

FIG. 2 is a diagram showing the files or tables in the application database of the present invention that are utilized for data storage and retrieval during the processing that values the categories of value within the organization;

FIG. 3 is a block diagram of an implementation of the present invention;

FIG. 4 is a diagram showing the data windows that are used for receiving information from and transmitting information to the user (20) during system processing;

FIG. 5A, FIG. 5B, FIG. 5C, FIG. 5D, FIG. 5E and FIG. 5F are block diagrams showing the sequence of steps in the present invention used for specifying system settings and for initializing and operating the data bots that extract, aggregate, store and manipulate information utilized in system processing from: user input, the basic financial system database, the operation management system database, the human resource information system database, external databases, the advanced financial system database, soft asset management system databases and the internet;

FIG. 6A, FIG. 6B and FIG. 6C are block diagrams showing the sequence of steps in the present invention that are utilized for initializing and operating the analysis bots;

FIG. 7 is a block diagram showing the sequence of steps in the present invention used for the analyzing enterprise market sentiment;

FIG. 8 is a block diagram showing the sequence of steps in the present invention used in trading organization stock and in preparing, displaying and optionally printing reports; and

FIG. 9 is a block diagram showing the sequence of steps in the present invention used for generating lists of value enhancing changes and calculating, displaying and optionally printing simulations of the effects of user-specified and/or system generated changes in business value drivers on the financial performance and the future value of the organization and the enterprises in the organization;

20250701 05404660

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG.1 provides an overview of the processing completed by the innovative system for business valuation. In accordance with the present invention, an automated method of and system (100) for business valuation is provided. Processing starts in this system (100) with a the specification of system settings and the initialization and activation of software data "bots" (200) that extract, aggregate, manipulate and store the data and user (20) input required for completing system processing. This information is extracted via a network (45) from a basic financial system database (5), an operation management system database (10), a human resource information system database (15), an external database (25), an advanced financial system database (30), soft asset management system databases (35) and the internet (40). These information extractions and aggregations may be influenced by a user (20) through interaction with a user-interface portion of the application software (700) that mediates the display, transmission and receipt of all information to and from a browser (800) that the user (20) interacts with. While only one database of each type (5, 10, 15, 25, 30 and 35) is shown in FIG. 1, it is to be understood that the system (100) can extract data from multiple databases of each type via the network (45). The preferred embodiment of the present invention contains a soft asset management system for each element of value being analyzed. Automating the extraction and analysis of data from each soft asset management system ensures that the management of each soft asset is considered and prioritized within the overall financial models for the organization and for each enterprise in the organization. It should also be understood that it is possible to complete a bulk extraction of data from each database (5, 10, 15, 25, 30 and 35) via the network (45) using data extraction applications such as Aclue from Decisionism and Power Center from Informatica before initializing the data bots. The data extracted in bulk could be stored in a single datamart or datawarehouse where the data bots could operate on the aggregated data.

All extracted information is stored in a file or table (hereinafter, table) within an application database (50) as shown in FIG. 2. The application database (50) contains tables for storing user input, extracted information and system calculations including a system settings table (140), a metadata mapping table (141), a conversion rules table (142), a basic financial system table (143), an operation system table (144), a human

resource system table (145), an external database table (146), an advanced finance system table (147), a soft asset system table (148), a bot date table (149), a keyword table (150), a classified text table (151), a geospatial measures table (152), a composite variables table (153), an industry ranking table (154), an element of value definition table (155), a component of value definition table (156), a cluster ID table (157), an element variables table (158), a vector table (159), a bot table (160), a cash flow table (161), a real option value table (162), an enterprise vector table (163), a report table (164), an equity purchase table (165), an enterprise sentiment table (166), a value driver change table (167), a simulation table (168) and a sentiment factors table (169). The application database (50) can optionally exist as a datamart, data warehouse or departmental warehouse. The system of the present invention has the ability to accept and store supplemental or primary data directly from user input, a data warehouse or other electronic files in addition to receiving data from the databases described previously. The system of the present invention also has the ability to complete the necessary calculations without receiving data from one or more of the specified databases. However, in the preferred embodiment all required information is obtained from the specified data sources (5, 10, 15, 25, 30, 35 and 40).

As shown in FIG. 3, the preferred embodiment of the present invention is a computer system (100) illustratively comprised of a user-interface personal computer (110) connected to an application server personal computer (120) via a network (45). The application server personal computer (120) is in turn connected via the network (45) to a database-server personal computer (130). The user interface personal computer (110) is also connected via the network (45) to an internet browser appliance (90) that contains browser software (800) such as Microsoft Internet Explorer or Netscape Navigator.

The database-server personal computer (130) has a read/write random access memory (131), a hard drive (132) for storage of the application database (50), a keyboard (133), a communications bus (134), a CRT display (135), a mouse (136), a CPU (137) and a printer (138).

The application-server personal computer (120) has a read/write random access memory (121), a hard drive (122) for storage of the non user interface portion of the application software (200, 300, 400, 500 and 600) of the present invention, a keyboard

(123), a communications bus (124), a CRT display (125), a mouse (126), a CPU (127) and a printer (128). While only one client personal computer is shown in FIG. 3, it is to be understood that the application-server personal computer (120) can be networked to fifty or more client personal computers (110) via the network (45). The application-server personal computer (120) can also be networked to fifty or more server, personal computers (130) via the network (45). It is to be understood that the diagram of FIG. 3 is merely illustrative of one embodiment of the present invention.

The user-interface personal computer (110) has a read/write random access memory (111), a hard drive (112) for storage of a client data-base (49) and the user-interface portion of the application software (700), a keyboard (113), a communications bus (114), a CRT display (115), a mouse (116), a CPU (117) and a printer (118).

The application software (200, 300, 400, 500, 600 and 700) controls the performance of the central processing unit (127) as it completes the calculations required to calculate the detailed business valuation. In the embodiment illustrated herein, the application software program (200, 300, 400, 500, 600 and 700) is written in a combination of C++ and Visual Basic®. The application software (200, 300, 400, 500, 600 and 700) can use Structured Query Language (SQL) for extracting data from the databases and the internet (5, 10, 15, 25, 30, 35 and 40). The user (20) can optionally interact with the user-interface portion of the application software (700) using the browser software (800) in the browser appliance (90) to provide information to the application software (200, 300, 400, 500, 600 and 700) for use in determining which data will be extracted and transferred to the application database (50) by the data bots.

User input is initially saved to the client database (49) before being transmitted to the communication bus (125) and on to the hard drive (122) of the application-server computer via the network (45). Following the program instructions of the application software, the central processing unit (127) accesses the extracted data and user input by retrieving it from the hard drive (122) using the random access memory (121) as computation workspace in a manner that is well known.

The computers (110, 120 and 130) shown in FIG. 3 illustratively are IBM PCs or clones or any of the more powerful computers or workstations that are widely available. Typical

memory configurations for client personal computers (110) used with the present invention should include at least 256 megabytes of semiconductor random access memory (111) and at least a 50 gigabyte hard drive (112). Typical memory configurations for the application-server personal computer (120) used with the present invention should include at least 1028 megabytes of semiconductor random access memory (121) and at least a 100 gigabyte hard drive (122). Typical memory configurations for the database-server personal computer (130) used with the present invention should include at least 2056 megabytes of semiconductor random access memory (135) and at least a 500 gigabyte hard drive (131).

Using the system described above, the value of the organization, each enterprise within the organization and each element of value can be broken down into the value categories listed in Table 1. As shown in Table 2 and Table 3, the value of the current-operation will be calculated using an income valuation. An integral part of most income valuation models is the calculation of the present value of the expected cash flows, income or profits associated with the current-operation. The present value of a stream of cash flows is calculated by discounting the cash flows at a rate that reflects the risk associated with realizing the cash flow. For example, the present value (PV) of a cash flow of ten dollars (\$10) per year for five (5) years would vary depending on the rate used for discounting future cash flows as shown below.

| Discount rate = 25% | | | | | | | | | | | | |
|---------------------|---|-----------|---|---------------------|---|---------------------|---|---------------------|---|---------------------|---|-------|
| PV | = | <u>10</u> | + | <u>10</u> | + | <u>10</u> | + | <u>10</u> | + | <u>10</u> | = | 26.89 |
| | | 1.25 | | (1.25) ² | | (1.25) ³ | | (1.25) ⁴ | | (1.25) ⁵ | | |

| Discount rate = 35% | | | | | | | | | | | | |
|---------------------|---|-----------|---|---------------------|---|---------------------|---|---------------------|---|---------------------|---|-------|
| PV | = | <u>10</u> | + | <u>10</u> | + | <u>10</u> | + | <u>10</u> | + | <u>10</u> | = | 22.20 |
| | | 1.35 | | (1.35) ² | | (1.35) ³ | | (1.35) ⁴ | | (1.35) ⁵ | | |

One of the first steps in evaluating the elements of current-operation value is extracting the data required to complete calculations in accordance with the formula that defines the value of the current-operation as shown in Table 6.

Table 6

| Value of current-operation = |
|---|
| (R) Value of forecast revenue from current-operation (positive) |
| + |
| (E) Value of forecast expense for current-operation (negative) |
| + |
| (C)* Value of current operation capital change forecast |

*Note: (C) can have a positive or negative value

The three components of current-operation value will be referred to as the revenue value (R), the expense value (E) and the capital value (C). Examination of the equation in Table 6 shows that there are three ways to increase the value of the current-operation - increase the revenue, decrease the expense or decrease the capital requirements (note: this statement ignores a fourth way to increase value - decrease interest rate used for discounting future cash flows).

In the preferred embodiment, the revenue, expense and capital requirement forecasts for the current operation, the real options and the contingent liabilities are obtained from an advanced financial planning system database (30) from an advanced financial planning system similar to the one disclosed in U.S. Patent 5,615,109. The extracted revenue, expense and capital requirement forecasts are used to calculate a cash flow for each period covered by the forecast for the organization and each enterprise in the organization by subtracting the expense and change in capital for each period from the revenue for each period. A steady state forecast for future periods is calculated after determining the steady state growth rate the best fits the calculated cash flow for the forecast time period. The steady state growth rate is used to calculate an extended cash flow forecast. The extended cash flow forecast is used to determine the Competitive Advantage Period (CAP) implicit in the enterprise market value.

While it is possible to use analysis bots to sub-divide each of the components of current operation value into a number of sub-components for analysis, the preferred embodiment

has a pre-determined number of sub-components for each component of value for the organization and each enterprise in the organization. The revenue value is not subdivided. In the preferred embodiment, the expense value is subdivided into five sub-components: the cost of raw materials, the cost of manufacture or delivery of service, the cost of selling, the cost of support and the cost of administration. The capital value is subdivided into six sub-components: cash, non-cash financial assets, production equipment, other assets (non financial, non production assets), financial liabilities and equity. The production equipment and equity sub-components are not used directly in evaluating the elements of value.

The components and sub-components of current-operation value will be used in calculating the value of: enterprise contribution, elements of value and sub-elements of value. Enterprise contribution will be defined as "the economic benefit that as a result of past transactions an enterprise is expected to provide to an organization." In a similar fashion, an element of value will be defined as "an identifiable entity or group of items that as a result of past transactions has provided and is expected to provide economic benefit to an enterprise". An item will be defined as a single member of the group that defines an element of value. For example, an individual salesman would be an "item" in the "element of value" sales staff. The data associated with performance of an individual item will be referred to as "item variables".

Analysis bots are used to determine enterprise and element of value lives and the percentage of: the revenue value, the expense value, and the capital value that are attributable to each element of value. The resulting values are then be added together to determine the valuation for different elements as shown by the example in Table 7.

2025-05-04 09:46:50

Table 7

| Gross Value | Percentage | Element Life/CAP | Net Value |
|-----------------------------|------------|------------------|-------------------|
| Revenue value = \$120M | 20% | 80% | Value = \$19.2 M |
| Expense value = (\$80M) | 10% | 100% | Value = (\$8.0) M |
| Capital value = (\$5M) | 5% | 80% | Value = (\$0.2) M |
| Total value = \$35M | | | |
| Net value for this element: | | | Value = \$11.0 M |

The valuation of an organization and the enterprises in the organization using the approach outlined above is completed in five distinct stages. As shown in FIG. 5A, FIG. 5B, FIG. 5C, FIG. 5D, FIG. 5E and FIG. 5F the first stage of processing (block 200 from FIG. 1) programs bots to continually extract, aggregate, manipulate and store the data from user input and databases and the internet (5, 10, 15, 25, 30, 35 or 40) as required for the analysis of business value. Bots are independent components of the application that have specific tasks to perform. As shown in FIG. 6A, FIG. 6B and FIG. 6C the second stage of processing (block 300 from FIG. 1) programs analysis bots to continually:

1. identify the item variables, item performance indicators and composite variables for each enterprise, element of value and sub-element of value that drive the components of value (revenue, expense and changes in capital),
2. create vectors that summarize the performance of the item variables and item performance indicators for each enterprise contribution, element of value and sub-element of value,
3. determine the appropriate cost of capital and value the organization and enterprise real options;
4. determine the appropriate cost of capital, value and allocate the industry real options to each organization or enterprise on the basis of relative element strength;
5. determine the expected life of each element of value and sub-element of value;
6. calculate the organization and enterprise current operation values and value the revenue, expense and capital components said current operations using the information prepared in the previous stage of processing;

7. specify and optimize predictive models to determine the relationship between the vectors determined in step 2 and the revenue, expense and capital values determined in step 6,
8. combine the results of the fifth, sixth and seventh stages of processing to determine the value of each, enterprise contribution, element and sub-element (as shown in Table 7);

The third stage of processing (block 400 from FIG. 1) analyzes the market sentiment associated with each enterprise as shown in FIG. 7. The fourth stage of processing (block 500 from FIG. 1) displays the results of the prior calculations in specified formats and optionally generates trades in enterprise stock as shown in FIG. 8. The fifth and final stage of processing (block 600 from FIG. 1) identifies potential improvements in organization and enterprise operation and analyzes the impact of proposed improvements on financial performance and business value for the organization and each enterprise as shown in FIG. 9.

SYSTEM SETTINGS AND DATA BOTS

The flow diagrams in FIG. 5A, FIG. 5B, FIG. 5C, FIG. 5D, FIG. 5E and FIG. 5F detail the processing that is completed by the portion of the application software (200) that extracts, aggregates, transforms and stores the information required for system operation from: the basic financial system database (5), operation management system database (10), human resource information system database (15), external database (25), advanced financial system database (30), soft asset management system database (35), the internet (40) and the user (20). A brief overview of the different databases will be presented before reviewing each step of processing completed by this portion (200) of the application software.

Corporate financial software systems are generally divided into two categories, basic and advanced. Advanced financial systems utilize information from the basic financial systems to perform financial analysis, financial planning and financial reporting functions. Virtually every commercial enterprise uses some type of basic financial system as they

are required to use these systems to maintain books and records for income tax purposes. An increasingly large percentage of these basic financial systems are resident in microcomputer and workstation systems. Basic financial systems include general-ledger accounting systems with associated accounts receivable, accounts payable, capital asset, inventory, invoicing, payroll and purchasing subsystems. These systems incorporate worksheets, files, tables and databases. These databases, tables and files contain information about the company operations and its related accounting transactions. As will be detailed below, these databases, tables and files are accessed by the application software of the present invention as required to extract the information required for completing a business valuation. The system is also capable of extracting the required information from a data warehouse (or datamart) when the required information has been pre-loaded into the warehouse.

General ledger accounting systems generally store only valid accounting transactions. As is well known, valid accounting transactions consist of a debit component and a credit component where the absolute value of the debit component is equal to the absolute value of the credit component. The debits and the credits are posted to the separate accounts maintained within the accounting system. Every basic accounting system has several different types of accounts. The effect that the posted debits and credits have on the different accounts depends on the account type as shown in Table 8.

Table 8

| Account Type: | Debit Impact: | Credit Impact: |
|----------------------|----------------------|-----------------------|
| Asset | Increase | Decrease |
| Revenue | Decrease | Increase |
| Expense | Increase | Decrease |
| Liability | Decrease | Increase |
| Equity | Decrease | Increase |

General ledger accounting systems also require that the asset account balances equal the sum of the liability account balances and equity account balances at all times.

The general ledger system generally maintains summary, dollar only transaction histories and balances for all accounts while the associated subsystems, accounts payable, accounts receivable, inventory, invoicing, payroll and purchasing, maintain more detailed historical transaction data and balances for their respective accounts. It is common practice for each subsystem to maintain the detailed information shown in Table 9 for each transaction.

Table 9

| Subsystem | Detailed Information |
|---------------------|---|
| Accounts Payable | Vendor, Item(s), Transaction Date, Amount Owed, Due Date, Account Number |
| Accounts Receivable | Customer, Transaction Date, Product Sold, Quantity, Price, Amount Due, Terms, Due Date, Account Number |
| Capital Assets | Asset ID, Asset Type, Date of Purchase, Purchase Price, Useful Life, Depreciation Schedule, Salvage Value |
| Inventory | Item Number, Transaction Date, Transaction Type, Transaction Qty, Location, Account Number |
| Invoicing | Customer Name, Transaction Date, Item(s) Sold, Amount Due, Due Date, Account Number |
| Payroll | Employee Name, Employee Title, Pay Frequency, Pay Rate, Account Number |
| Purchasing | Vendor, Item(s), Purchase Quantity, Purchase Price(s), Due Date, Account Number |

As is well known, the output from a general ledger system includes income statements, balance sheets and cash flow statements in well defined formats which assist management in measuring the financial performance of the firm during the prior periods when data input and system processing have been completed.

While basic financial systems are similar between firms, operation management systems vary widely depending on the type of company they are supporting. These systems typically have the ability to not only track historical transactions but to forecast

future performance. For manufacturing firms, operation management systems such as Enterprise Resource Planning Systems (ERP), Material Requirement Planning Systems (MRP), Purchasing Systems, Scheduling Systems and Quality Control Systems are used to monitor, coordinate, track and plan the transformation of materials and labor into products. Systems similar to the one described above may also be useful for distributors to use in monitoring the flow of products from a manufacturer.

Operation Management Systems in manufacturing firms may also monitor information relating to the production rates and the performance of individual production workers, production lines, work centers, production teams and pieces of production equipment including the information shown in Table 10.

Table 10

| Operation Management System – Production Information | |
|---|---|
| 1. ID number (employee id/machine id) | 10. Cumulative training time |
| 2. Actual hours – last batch | 11. Job(s) certifications |
| 3. Standard hours – last batch | 12. Actual scrap - last batch |
| 4. Actual hours – year to date | 13. Scrap allowance - last batch |
| 5. Actual/Standard hours - year to date % | 14. Actual scrap/allowance - year to date |
| 6. Actual setup time - last batch | 15. Rework time/unit last batch |
| 7. Standard setup time - last batch | 16. Rework time/unit year to date |
| 8. Actual setup hours - year to date | 17. QC rejection rate - batch |
| 9. Actual/Standard setup hrs - yr to date % | 18. QC rejection rate - year to date |

Operation management systems are also useful for tracking requests for service to repair equipment in the field or in a centralized repair facility. Such systems generally store information similar to that shown below in Table 11.

Table 11

| Operation Management System - Service Call Information | |
|---|------------------------------------|
| 1. Customer name | 11. Promised type of response |
| 2. Customer number | 12. Time person dispatched to call |
| 3. Contract number | 13. Name of person handling call |
| 4. Service call number | 14. Time of arrival on site |
| 5. Time call received | 15. Time of repair completion |
| 6. Product(s) being fixed | 16. Actual response type |
| 7. Serial number of equipment | 17. Part(s) replaced |
| 8. Name of person placing call | 18. Part(s) repaired |
| 9. Name of person accepting call | 19. 2nd call required |
| 10. Promised response time | 20. 2nd call number |

Computer based human resource systems may some times be packaged or bundled within enterprise resource planning systems such as those available from SAP, Oracle and Peoplesoft. Human resource systems are increasingly used for storing and maintaining corporate records concerning active employees in sales, operations and the other functional specialties that exist within a modern corporation. Storing records in a centralized system facilitates timely, accurate reporting of overall manpower statistics to the corporate management groups and the various government agencies that require periodic updates. In some cases human resource systems include the company payroll system as a subsystem. In the preferred embodiment of the present invention, the payroll system is part of the basic financial system. These systems can also be used for detailed planning regarding future manpower requirements. Human resource systems typically incorporate worksheets, files, tables and databases that contain information about the current and future employees. As will be detailed below, these databases, tables and files are accessed by the application software of the present invention as required to extract the information required for completing a business valuation. It is common practice for human resource systems to store the information shown in Table 12 for each employee.

Table 12

| Human Resource System Information | |
|--|---------------------------------------|
| 1. Employee name | 10. Employee start date - company |
| 2. Job title | 11. Employee start date - department |
| 3. Job code | 12. Employee start date - current job |
| 4. Rating | 13. Training courses completed |
| 5. Division | 14. Cumulative training expenditures |
| 6. Department | 15. Salary history |
| 7. Employee No./(Social Security Number) | 16. Current salary |
| 8. Year to date - hours paid | 17. Educational background |
| 9. Year to date - hours worked | 18. Current supervisor |

External databases can be used for obtaining information that enables the definition and evaluation of a variety of things including elements of value, sentiment factors, industry real options and composite variables. In some cases information from these databases can be used to supplement information obtained from the other databases and the internet (5, 10, 15, 30, 35 and 40). In the system of the present invention, the information extracted from external databases (25) can be in the forms listed in Table 13.

| Types of information |
|---|
| a) numeric information such as that found in the SEC Edgar database and the databases of financial infomediaries such as FirstCall, IBES and Compustat, |
| b) text information such as that found in the Lexis Nexis database and databases containing past issues from specific publications, |
| c) multimedia information such as video and audio clips, and |
| d) geospatial data. |

The system of the present invention uses different "bot" types to process each distinct data type from external databases (25). The same "bot types" are also used for extracting each of the different types of data from the internet (40). The system of the

present invention must have access to at least one external database (25) that provides information regarding the equity prices for each enterprise in the organization and the equity prices and financial performance of competitors.

Advanced financial systems may also use information from external databases (25) and the internet (40) in completing their processing. Advanced financial systems include financial planning systems and activity based costing systems. Activity based costing systems may be used to supplement or displace the operation of the expense component analysis segment of the present invention as disclosed previously. Financial planning systems generally use the same format used by basic financial systems in forecasting income statements, balance sheets and cash flow statements for future periods. Management uses the output from financial planning systems to highlight future financial difficulties with a lead time sufficient to permit effective corrective action and to identify problems in company operations that may be reducing the profitability of the business below desired levels. These systems are most often developed by individuals within companies using 2 and 3 dimensional spreadsheets such as Lotus 1-2-3®, Microsoft Excel® and Quattro Pro®. In some cases, financial planning systems are built within an executive information system (EIS) or decision support system (DSS). For the preferred embodiment of the present invention, the advanced financial system database is similar to the financial planning system database detailed in U.S. Patent 5,165,109 for "Method of and System for Generating Feasible, Profit Maximizing Requisition Sets", by Jeff S. Eder, the disclosure of which is incorporated herein by reference.

While advanced financial planning systems have been around for some time, soft asset management systems are a relatively recent development. Their appearance is further proof of the increasing importance of "soft" assets. Soft asset management systems include: alliance management systems, brand management systems, customer relationship management systems, channel management systems, intellectual property management systems, process management systems and vendor management systems. Soft asset management systems are similar to operation management systems in that they generally have the ability to forecast future events as well as track historical occurrences. Customer relationship management systems are the most well established soft asset management systems at this point and will be the focus of the discussion

regarding soft asset management system data. In firms that sell customized products, the customer relationship management system is generally integrated with an estimating system that tracks the flow of estimates into quotations, orders and eventually bills of lading and invoices. In other firms that sell more standardized products, customer relationship management systems generally are used to track the sales process from lead generation to lead qualification to sales call to proposal to acceptance (or rejection) and delivery. All customer relationship management systems would be expected to track all of the customer's interactions with the enterprise after the first sale and store information similar to that shown below in Table 14.

Table 14

| Customer Relationship Management System – Information | |
|--|--------------------------------------|
| 1. Customer/Potential customer name | 9. Sales call history |
| 2. Customer number | 10. Sales contact history |
| 3. Address | 11. Sales history: product/qty/price |
| 4. Phone number | 12. Quotations: product/qty/price |
| 5. Source of lead | 13. Custom product percentage |
| 6. Date of first purchase | 14. Payment history |
| 7. Date of last purchase | 15. Current A/R balance |
| 8. Last sales call/contact | 16. Average days to pay |

System processing of the information from the different databases and the internet (5, 10, 15, 25, 30, 35 and 40) described above starts in a block 201, FIG. 5A, which immediately passes processing to a software block 202. The software in block 202 prompts the user (20) via the system settings data window (701) to provide system setting information. The system setting information entered by the user (20) is transmitted via the network (45) back to the application server (120) where it is stored in the system settings table (140) in the application database (50) in a manner that is well known. The specific inputs the user (20) is asked to provide at this point in processing are shown in Table 15.

Table 15

1. New run or structure revision?
2. Continuous, If yes, frequency? (hourly, daily, weekly, monthly or quarterly)
3. Structure of virtual organization (organization, enterprises and sub-elements)
4. Organization checklist
5. Enterprise checklist
6. Base account structure
7. Metadata standard (XML, MS OIM, MDC)
8. Location of basic financial system database and metadata
9. Location of advanced financial system database and metadata
10. Location of human resource information system database and metadata
11. Location of operation management system database and metadata
12. Location of soft asset management system databases and metadata
13. Location of external database and metadata
14. Location of account structure
15. Base currency
16. Location of database and metadata for equity information
17. Location of database and metadata for debt information
18. Location of database and metadata for tax rate information
19. Location of database and metadata for currency conversion rate information
20. Geospatial data? If yes, identity of geocoding service.
21. The maximum number of generations to be processed without improving fitness
22. Default clustering algorithm (selected from list) and maximum cluster number
23. Amount of cash and marketable securities required for day to day operations
24. Weighted average cost of capital (optional input)
25. Number of months a product is considered new after it is first produced
26. Organization industry segments (SIC Code)
27. Enterprise industry segments (SIC Code)
28. Primary competitors by industry segment
29. Management report types (text, graphic, both)
30. Default reports
31. Trading in enterprise equity authorized?

- 32. On-line equity trading account information
- 33. Default Missing Data Procedure
- 34. Maximum time to wait for user input

The organization and enterprise checklists are used by a "rules" engine (such as the one available from Neuron Data) in block 202 to influence the number and type of items with pre-defined metadata mapping for each category of value. For example, if the checklists indicate that the organization and enterprises are focused on branded, consumer markets, then additional brand related factors will be pre-defined for mapping. The application of these system settings will be further explained as part of the detailed explanation of the system operation.

The software in block 202 also uses the current system date to determine the time periods (months) that require data in order to complete the current operation and the real option valuations and stores the resulting date range in the system settings table (140). In the preferred embodiment the valuation of the current operation by the system utilizes basic finance, advanced financial, soft asset management, external database and human resource data for the three year period before and the three year forecast period after the current date.

After the storage of system setting data is complete, processing advances to a software block 203. The software in block 203 prompts the user (20) via the metadata and conversion rules window (702) to map metadata using the standard specified by the user (20) (XML, Microsoft's Open Information Model of the Metadata Coalitions specification) from the basic financial system database (5), the operation management system database (10), the human resource information system database (15), the external database (25), the advanced financial system database (30) and the soft asset management system database (35) to the organizational hierarchy stored in the system settings table (140) and to the pre-specified fields in the metadata mapping table (141). Pre-specified fields in the metadata mapping table include, the revenue, expense and capital components and sub-components for the organization and each enterprise and pre-specified fields for expected value drivers. Because the bulk of the information being extracted is financial information, the metadata mapping often takes the form of

specifying the account number ranges that correspond to the different fields in the metadata mapping table (141). Table 16, shows the base account number structure that the account numbers in the other systems must align with. For example, using the structure shown below, the revenue component for the organization could be specified as organization 01, any enterprise number, any department number, accounts 400 to 499 (the revenue account range) with any sub-account.

Table 16

| Account Number | 01 - | 800 - | 901 - | 677- | 003 |
|----------------|--------------|-------------|------------|---------|-------------|
| Segment | Organization | Enterprise | Department | Account | Sub-account |
| Subgroup | Products | Workstation | Marketing | Labor | P.R. |
| Position | 5 | 4 | 3 | 2 | 1 |

As part of the metadata mapping process, any database fields that are not mapped to pre-specified fields are defined by the user (20) as component of value. elements of value or non-relevant attributes and "mapped" in the metadata mapping table (141) to the corresponding fields in each database in a manner identical to that described above for the pre-specified fields. After all fields have been mapped to the metadata mapping table (141), the software in block 203 prompts the user (20) via the metadata and conversion rules window (702) to provide conversion rules for each metadata field for each data source. Conversion rules will include information regarding currency conversions and conversion for units of measure that may be required to accurately and consistently analyze the data. The inputs from the user (20) regarding conversion rules are stored in the conversion rules table (142) in the application database. When conversion rules have been stored for all fields from every data source, then processing advances to a software block 204.

The software in block 204 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change then processing

advances to a software block 212. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 207.

The software in block 207 checks the bot date table (149) and deactivates any basic financial system data bots with creation dates before the current system date and retrieves information from the system setting table (140), metadata mapping table (141) and conversion rules table (142). The software in block 207 then initializes data bots for each field in the metadata mapping table (141) that mapped to the basic financial system database (5) in accordance with the frequency specified by user (20) in the system settings table (140). Bots are independent components of the application that have specific tasks to perform. In the case of data acquisition bots, their tasks are to extract and convert data from a specified source and then store it in a specified location. Each data bot initialized by software block 207 will store its data in the basic financial system table (143). Every data acquisition bot for every data source contains the information shown in Table 17.

Table 17

| |
|--|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. The data source location |
| 3. Mapping information |
| 4. Timing of extraction |
| 5. Conversion rules (if any) |
| 6. Storage Location (to allow for tracking of source and destination events) |
| 7. Creation date (day, hour, minute, second) |

After the software in block 207 initializes all the bots for the basic financial system database, processing advances to a block 208. In block 208, the bots extract and convert data in accordance with their preprogrammed instructions in accordance with the frequency specified by user (20) in the system settings table (140). As each bot extracts and converts data from the basic financial system database (5), processing advances to a software block 209 before the bot completes data storage. The software in block 209 checks the basic financial system metadata to see if all fields have been extracted. If the software in block 209 finds no unmapped data fields, then the extracted, converted data

is stored in the basic financial system table (143). Alternatively, if there are fields that haven't been extracted, then processing advances to a block 210. The software in block 210 prompts the user (20) via the metadata and conversion rules window (702) to provide metadata and conversion rules for each new field. The information regarding the new metadata and conversion rules is stored in the metadata mapping table (141) and conversion rules table (142) while the extracted, converted data is stored in the basic financial system table (143). It is worth noting at this point that the activation and operation of bots that don't have unmapped fields continues. Only bots with unmapped fields "wait" for user input before completing data storage. The new metadata and conversion rule information will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing passes, on to software block 212.

The software in block 212 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change then processing advances to a software block 224. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 221.

The software in block 221 checks the bot date table (149) and deactivates any operations management system data bots with creation dates before the current system date and retrieves information from the system setting table (140), metadata mapping table (141) and conversion rules table (142). The software in block 221 then initializes data bots for each field in the metadata mapping table (141) that mapped to the operations management system database (10) in accordance with the frequency specified by user (20) in the system settings table (140). Each data bot initialized by software block 221 will store its data in the operations system table (144).

After the software in block 221 initializes all the bots for the operations management system database, processing advances to a block 222. In block 222, the bots extract and convert data in accordance with their preprogrammed instructions in accordance with the frequency specified by user (20) in the system settings table (140). As each bot extracts and converts data from the operations management system database (10), processing advances to a software block 209 before the bot completes data storage. The software in

block 209 checks the operations management system metadata to see if all fields have been extracted. If the software in block 209 finds no unmapped data fields, then the extracted, converted data is stored in the operations system table (144). Alternatively, if there are fields that haven't been extracted, then processing advances to a block 210. The software in block 210 prompts the user (20) via the metadata and conversion rules window (702) to provide metadata and conversion rules for each new field. The information regarding the new metadata and conversion rules is stored in the metadata mapping table (141) and conversion rules table (142) while the extracted, converted data is stored in the operations system table (144). It is worth noting at this point that the activation and operation of bots that don't have unmapped fields continues. Only bots with unmapped fields "wait" for user input before completing data storage. The new metadata and conversion rule information will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes, on to software block 224.

The software in block 224 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change then processing advances to a software block 228. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 225.

The software in block 225 checks the bot date table (149) and deactivates any human resource management system data bots with creation dates before the current system date and retrieves information from the system setting table (140), metadata mapping table (141) and conversion rules table (142). The software in block 225 then initializes data bots for each field in the metadata mapping table (141) that mapped to the human resource management system database (15) in accordance with the frequency specified by user (20) in the system settings table (140). Each data bot initialized by software block 225 will store its data in the human resource system table (145).

After the software in block 225 initializes all the bots for the human resource management system database, processing advances to a block 226. In block 226, the bots extract and convert data in accordance with their preprogrammed instructions in accordance with the frequency specified by user (20) in the system settings table (140).

As each bot extracts and converts data from the human resource management system database (15), processing advances to a software block 209 before the bot completes data storage. The software in block 209 checks the human resource management system metadata to see if all fields have been extracted. If the software in block 209 finds no unmapped data fields, then the extracted, converted data is stored in the human resource system table (145). Alternatively, if there are fields that haven't been extracted, then processing advances to a block 210. The software in block 210 prompts the user (20) via the metadata and conversion rules window (702) to provide metadata and conversion rules for each new field. The information regarding the new metadata and conversion rules is stored in the metadata mapping table (141) and conversion rules table (142) while the extracted, converted data is stored in the human resource system table (145). It is worth noting at this point that the activation and operation of bots that don't have unmapped fields continues. Only bots with unmapped fields "wait" for user input before completing data storage. The new metadata and conversion rule information will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes, on to software block 228.

The software in block 228 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change then processing advances to a software block 244. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 241.

The software in block 241 checks the bot date table (149) and deactivates any external database data bots with creation dates before the current system date and retrieves information from the system setting table (140), metadata mapping table (141) and conversion rules table (142). The software in block 241 then initializes data bots for each field in the metadata mapping table (141) that mapped to the external database (25) in accordance with the frequency specified by user (20) in the system settings table (140). Each data bot initialized by software block 241 will store its data in the external database table (146).

After the software in block 241 initializes all the bots for the external database, processing advances to a block 242. In block 242, the bots extract and convert data in

accordance with their preprogrammed instructions. As each bot extracts and converts data from the external database (25), processing advances to a software block 209 before the bot completes data storage. The software in block 209 checks the external database metadata to see if all fields have been extracted. If the software in block 209 finds no unmapped data fields, then the extracted, converted data is stored in the external database table (146). Alternatively, if there are fields that haven't been extracted, then processing advances to a block 210. The software in block 210 prompts the user (20) via the metadata and conversion rules window (702) to provide metadata and conversion rules for each new field. The information regarding the new metadata and conversion rules is stored in the metadata mapping table (141) and conversion rules table (142) while the extracted, converted data is stored in the external database table (146). It is worth noting at this point that the activation and operation of bots that don't have unmapped fields continues. Only bots with unmapped fields "wait" for user input before completing data storage. The new metadata and conversion rule information will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes, on to software block 244.

The software in block 244 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change then processing advances to a software block 248. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 245.

The software in block 245 checks the bot date table (149) and deactivates any advanced financial system data bots with creation dates before the current system date and retrieves information from the system setting table (140), metadata mapping table (141) and conversion rules table (142). The software in block 245 then initializes data bots for each field in the metadata mapping table (141) that mapped to the advanced financial system database (30) in accordance with the frequency specified by user (20) in the system settings table (140). Each data bot initialized by software block 245 will store its data in the advanced financial system database table (147).

After the software in block 245 initializes all the bots for the advanced financial system database, processing advances to a block 246. In block 246, the bots extract and convert

data in accordance with their preprogrammed instructions in accordance with the frequency specified by user (20) in the system settings table (140). As each bot extracts and converts data from the advanced financial system database (30), processing advances to a software block 209 before the bot completes data storage. The software in block 209 checks the advanced financial system database metadata to see if all fields have been extracted. If the software in block 209 finds no unmapped data fields, then the extracted, converted data is stored in the advanced financial system database table (147). Alternatively, if there are fields that haven't been extracted, then processing advances to a block 210. The software in block 210 prompts the user (20) via the metadata and conversion rules window (702) to provide metadata and conversion rules for each new field. The information regarding the new metadata and conversion rules is stored in the metadata mapping table (141) and conversion rules table (142) while the extracted, converted data is stored in the advanced financial system database table (147). It is worth noting at this point that the activation and operation of bots that don't have unmapped fields continues. Only bots with unmapped fields "wait" for user input before completing data storage. The new metadata and conversion rule information will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes, on to software block 248.

The software in block 248 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change then processing advances to a software block 264. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 261.

The software in block 261 checks the bot date table (149) and deactivates any soft asset management system data bots with creation dates before the current system date and retrieves information from the system setting table (140), metadata mapping table (141) and conversion rules table (142). The software in block 261 then initializes data bots for each field in the metadata mapping table (141) that mapped to a soft asset management system database (35) in accordance with the frequency specified by user (20) in the system settings table (140). Extracting data from each soft asset management system ensures that the management of each soft asset is considered and prioritized

within the overall financial models for the organization and each enterprise in the organization. Each data bot initialized by software block 261 will store its data in the soft asset system table (148).

After the software in block 261 initializes bots for all soft asset management system databases, processing advances to a block 262. In block 262, the bots extract and convert data in accordance with their preprogrammed instructions in accordance with the frequency specified by user (20) in the system settings table (140). As each bot extracts and converts data from the soft asset management system databases (35), processing advances to a software block 209 before the bot completes data storage. The software in block 209 checks the metadata for the soft asset management system databases to see if all fields have been extracted. If the software in block 209 finds no unmapped data fields, then the extracted, converted data is stored in the soft asset system table (148). Alternatively, if there are fields that haven't been extracted, then processing advances to a block 210. The software in block 210 prompts the user (20) via the metadata and conversion rules window (702) to provide metadata and conversion rules for each new field. The information regarding the new metadata and conversion rules is stored in the metadata mapping table (141) and conversion rules table (142) while the extracted, converted data is stored in the soft asset system table (148). It is worth noting at this point that the activation and operation of bots that don't have unmapped fields continues. Only bots with unmapped fields "wait" for user input before completing data storage. The new metadata and conversion rule information will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes, on to software block 264.

The software in block 264 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change then processing advances to a software block 276. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 265.

The software in block 265 prompts the user (20) via the identification and classification rules window (703) to identify keywords such as company names, brands, trademarks, competitors for pre-specified fields in the metadata mapping table (141). The user (20)

also has the option of mapping keywords to other fields in the metadata mapping table (141). After specifying the keywords, the user (20) is prompted to select and classify descriptive terms for each keyword. The input from the user (20) is stored in the keyword table (150) in the application database before processing advances to a software block 266.

The software in block 266 checks the bot date table (149) and deactivates any internet text bots with creation dates before the current system date and retrieves information from the system settings table (140), the metadata mapping table (141) and the keyword table (150). The software in block 266 then initializes internet text bots for each field in the metadata mapping table (141) that mapped to a keyword in accordance with the frequency specified by user (20) in the system settings table (140) before advancing processing to a software block 267.

Bots are independent components of the application that have specific tasks to perform. In the case of text bots, their tasks are to locate, count and classify keyword matches from a specified source and then store their findings in a specified location. Each text bot initialized by software block 266 will store the location, count and classification data it discovers in the classified text table (151). Multimedia data can be processed using bots with essentially the same specifications if software to translate and parse the multimedia content is included in each bot. Every internet text bot contains the information shown in Table 18.

Table 18

| |
|---|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. Creation date (day, hour, minute, second) |
| 3. Storage location |
| 4. Mapping information |
| 5. Home URL |
| 6. Keyword |
| 7. Descriptive term 1 |
| To |
| 7+n. Descriptive term n |

In block 267 the text bots locate and classify data from the external database (25) in accordance with their programmed instructions in accordance with the frequency specified by user (20) in the system settings table (140). As each text bot locates and classifies data from the internet (40) processing advances to a software block 268 before the bot completes data storage. The software in block 268 checks to see if all keyword hits are associated with descriptive terms that have been been classified. If the software in block 268 doesn't find any unclassified "hits", then the address, count and classified text are stored in the classified text table (151). Alternatively, if there are terms that haven't been classified, then processing advances to a block 269. The software in block 269 prompts the user (20) via the identification and classification rules window (703) to provide classification rules for each new term. The information regarding the new classification rules is stored in the keyword table (150) while the newly classified text is stored in the classified text table (151). It is worth noting at this point that the activation and operation of bots that don't have unclassified fields continues. Only bots with unclassified fields will "wait" for user input before completing data storage. The new classification rules will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes, on to a software block 270.

The software in block 270 checks the bot date table (149) and deactivates any external database text bots with creation dates before the current system date and retrieves information from the system settings table (140), the metadata mapping table (141) and the keyword table (150). The software in block 270 then initializes external database text bots for each field in the metadata mapping table (141) that mapped to a keyword in accordance with the frequency specified by user (20) in the system settings table (140) before advancing processing to a software block 271. Every text bot initialized by software block 270 will store the location, count and classification data it discovers in the classified text table (151). Every external database text bot contains the information shown in Table 19.

Table 19

| |
|---|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. Creation date (day, hour, minute, second) |
| 3. Storage location |
| 4. Mapping information |
| 5. Data Source |
| 6. Keyword |
| 7. Descriptive term 1 |
| To |
| 7+n. Descriptive term n |

In block 271 the text bots locate and classify data from the external database (25) in accordance with its programmed instructions with the frequency specified by user (20) in the system settings table (140). As each text bot locates and classifies data from the external database (25) processing advances to a software block 268 before the bot completes data storage. The software in block 268 checks to see if all keyword hits are associated with descriptive terms that have been been classified. If the software in block 268 doesn't find any unclassified "hits", then the address, count and classified text are stored in the classified text table (151). Alternatively, if there are terms that haven't been classified, then processing advances to a block 269. The software in block 269 prompts

the user (20) via the identification and classification rules window (703) to provide classification rules for each new term. The information regarding the new classification rules is stored in the keyword table (150) while the newly classified text is stored in the classified text table (151). It is worth noting at this point that the activation and operation of bots that don't have unclassified fields continues. Only bots with unclassified fields "wait" for user input before completing data storage. The new classification rules will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes, on to software block 276.

The software in block 276 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change then processing advances to a software block 280. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 277.

The software in block 277 checks the system setting table (140) to see if there is geocoded data in the application database (50) and to determine which on-line geocoding service (Centrus™ from QM Soft or MapMarker™ from MapInfo) is being used. If geospatial data is not being used, then processing advances to a block 291. Alternatively, if the software in block 277 determines that geospatial data is being used, processing advances to a software block 278.

The software in block 278 prompts the user (20) via the geospatial measure definitions window (709) to define the measures that will be used in evaluating the elements of value. After specifying the measures, the user (20) is prompted to select the geospatial locus for each measure from the data already stored in the application database (50). The input from the user (20) is stored in the geospatial measures table (152) in the application database before processing advances to a software block 279.

The software in block 279 checks the bot date table (149) and deactivates any geospatial bots with creation dates before the current system date and retrieves information from the system settings table (140), the metadata mapping table (141) and the geospatial measures table (152). The software in block 279 then initializes geospatial bots for each field in the metadata mapping table (141) that mapped to geospatial data in

the application database (50) in accordance with the frequency specified by user (20) in the system settings table (140) before advancing processing to a software block 280.

Bots are independent components of the application that have specific tasks to perform. In the case of geospatial bots, their tasks are to calculate user specified measures using a specified geocoding service and then store the measures in a specified location. Each geospatial bot initialized by software block 279 will store the measures it calculates in the application database table where the geospatial data was found. Tables that could include geospatial data include: the basic financial system table (143), the operation system table (144), the human resource system table (145), the external database table (146), the advanced finance system table (147) and the soft asset system table (148). Every geospatial bot contains the information shown in Table 20.

Table 20

| |
|---|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. Creation date (day, hour, minute, second) |
| 3. Mapping information |
| 4. Storage location |
| 5. Geospatial locus |
| 6. Geospatial measure |
| 7. Geocoding service |

In block 280 the geospatial bots locate data and complete measurements in accordance with their programmed instructions with the frequency specified by the user (20) in the system settings table (140). As each geospatial bot retrieves data and calculates the geospatial measures that have been specified, processing advances to a block 281 before the bot completes data storage. The software in block 281 checks to see if all geospatial data located by the bot has been measured. If the software in block 281 doesn't find any unmeasured data, then the measurement is stored in the application database (50). Alternatively, if there are data elements that haven't been measured, then processing advances to a block 282. The software in block 282 prompts the user (20) via the geospatial measure definition window (709) to provide measurement

rules for each new term. The information regarding the new measurement rules is stored in the geospatial measures table (152) while the newly calculated measurement is stored in the appropriate table in the application database (50). It is worth noting at this point that the activation and operation of bots that don't have unmeasured fields continues. Only the bots with unmeasured fields "wait" for user input before completing data storage. The new measurement rules will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes on to a software block 291.

The software in block 291 checks: the basic financial system table (143), the operation system table (144), the human resource system table (145), the external database table (146), the advanced finance system table (147), the soft asset system table (148), the classified text table (151) and the geospatial measures table (152) to see if data is missing from any of the periods required for system calculation. The range of required dates was previously calculated by the software in block 202. If there is no data missing from any period, then processing advances to a software block 293. Alternatively, if there is missing data for any field for any period, then processing advances to a block 292.

The software in block 292, prompts the user (20) via the missing data window (704) to specify the method to be used for filling the blanks for each item that is missing data. Options the user (20) can choose from for filling the blanks include: the average value for the item over the entire time period, the average value for the item over a specified period, zero, the average of the preceeding item and the following item values and direct user input for each missing item. If the user (20) doesn't provide input within a specified interval, then the default missing data procedure specified in the system settings table (140) is used. When all the blanks have been filled and stored for all of the missing data, system processing advances to a block 293.

The software in block 293 calculates attributes by item for each numeric data field in the basic financial system table (143), the operation system table (144), the human resource system table (145), the external database table (146), the advanced finance system table (147) and the soft asset system table (148). The attributes calculated in this step include: cumulative total value, the period to period rate of change in value, the rolling average value and a series of time lagged values. In a similar fashion the software

in block 293 calculates attributes for each date field in the specified tables including time since last occurrence, cumulative time since first occurrence, average frequency of occurrence and the rolling average frequency of occurrence. The numbers derived from numeric and date fields are collectively referred to as "item performance indicators". The software in block 293 also calculates pre-specified combinations of variables called composite variables for measuring the strength of the different elements of value. The item performance indicators are stored in the table where the item source data was obtained and the composite variables are stored in the composite variables table (153) before processing advances to a block 294.

The software in block 294 uses attribute derivation algorithms such as the AQ program to create combinations of the variables that weren't pre-specified for combination. While the AQ program is used in the preferred embodiment of the present invention, other attribute derivation algorithms such as the LINUS algorithms, may be used to the same effect. The software creates these attributes using both item variables that were specified as "element" variables and item variables that were not. The resulting composite variables are stored in the composite variables table (153) before processing advances to a block 295.

The software in block 295 uses Data Envelopment Analysis (hereinafter, DEA) to determine the relative industry ranking of the organization and enterprises being examined using the composite variables calculated in block 293. For example, DEA can be used to determine the relative efficiency of a company in receiving favorable press mentions per dollar spent on advertising. When all pre-specified industry rankings have been calculated and stored in the industry ranking table (154), processing advances to a software block 296.

The software in block 296 uses pattern-matching algorithms to assign pre-designated data fields for different elements of value to pre-defined groups with numerical values. This type of analysis is useful in classifying purchasing patterns and/or communications patterns as "heavy", "light", "moderate" or "sporadic". The assignments are calculated using the "rolling average" value for each field. The classification and the numeric value associated with the classification are stored in the application database (50) table where the data field is located before processing advances to a block 297.

The software in block 297 retrieves data from the metadata mapping table (141), creates and then stores the definitions for the pre-defined components of value in the components of value definition table (155). As discussed previously, the revenue component of value is not divided into sub-components, the expense value is divided into five sub-components (the cost of raw materials, the cost of manufacture or delivery of service, the cost of selling, the cost of support and the cost of administration) and the capital value is divided into six sub-components: (cash, non-cash financial assets, production equipment, other assets, financial liabilities and equity) in the preferred embodiment. When data storage is complete, processing advances to a software block 302 to begin the analysis of the extracted data using analysis bots.

ANALYSIS BOTS

The flow diagrams in FIG. 6A, FIG. 6B and FIG. 6C detail the processing that is completed by the portion of the application software (300) that programs analysis bots to:

1. identify the item variables, item performance indicators and composite variables for each enterprise, element of value and sub-element of value that drive the components of value (revenue, expense and changes in capital),
2. create vectors that summarize the performance of the item variables and item performance indicators for each enterprise contribution, element of value and sub-element of value,
3. determine the appropriate cost of capital and value the organization and enterprise real options;
4. determine the appropriate cost of capital, value and allocate the industry real options to each organization or enterprise on the basis of relative element strength;
5. determine the expected life of each element of value and sub-element of value;
6. calculate the organization and enterprise current operation values and value the revenue, expense and capital components said current operations using the information prepared in the previous stage of processing;
7. specify and optimize predictive models to determine the relationship between the vectors determined in step 2 and the revenue, expense and capital values determined in step 6,

8. combine the results of the fifth, sixth and seventh stages of processing to determine the value of each, enterprise contribution, element and sub-element (as shown in Table 7);

Processing in this portion of the application begins in software block 302. The software in block 302 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change then processing advances to a software block 3110. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 303.

The software in block 303 retrieves data from the meta data mapping table (141) and the soft asset system table (148) and then assigns item variables, item performance indicators and composite variables to each element of value using a two step process. First, item variables and item performance indicators are assigned to elements of value based on the soft asset management system they correspond to (for example, all item variables from a brand management system and all item performance indicators derived from brand management system variables are assigned to the brand element of value). Second, pre-defined composite variables are assigned to the element of value they were assigned to measure in the metadata mapping table (141). After the assignment of variables and indicators to elements is complete, the resulting assignments are saved to the element of value definition table (155) and processing advances to a block 304.

The software in block 304 checks the bot date table (149) and deactivates any clustering bots with creation dates before the current system date. The software in block 304 then initializes bots as required for each component of value. The bots: activate in accordance with the frequency specified by the user (20) in the system settings table (140), retrieve the information from the system settings table (140), the metadata mapping table (141) and the component of value definition table (156) as required and define segments for the component of value data before saving the resulting cluster information in the application database (50).

Bots are independent components of the application that have specific tasks to perform. In the case of predictive model bots, their primary task is to segment the

component and sub-component of value variables into distinct clusters that share similar characteristics. The clustering bot assigns a unique id number to each "cluster" it identifies and stores the unique id numbers in the cluster id table (157). Every item variable for every component and sub-component of value is assigned to one of the unique clusters. The cluster id for each variable is saved in the data record for each item variable in the table where it resides. The item variables are segmented into a number of clusters less than or equal to the maximum specified by the user (20) in the system settings. The data is segmented using the "default" clustering algorithm the user (20) specified in the system settings. The system of the present invention provides the user (20) with the choice of several clustering algorithms including: an unsupervised "Kohonen" neural network, K-nearest neighbor, Expectation Maximization (EM) and the segmental K-means algorithm. For algorithms that normally require the number of clusters to be specified the bot will iterate the number of clusters until it finds the cleanest segmentation for the data. Every clustering bot contains the information shown in Table 21.

Table 21

| |
|---|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. Creation date (day, hour, minute, second) |
| 3. Mapping information |
| 4. Storage location |
| 5. Component or subcomponent of value |
| 6. Clustering algorithm type |
| 7. Maximum number of clusters |
| 8. Variable 1 |
| ... |
| 8+n. Variable n |

When bots in block 304 have identified and stored cluster assignments for the item variables associated with each component and subcomponent of value, processing advances to a software block 305.

The software in block 305 checks the bot date table (149) and deactivates any predictive model bots with creation dates before the current system date. The software in block 305 then retrieves the information from the system settings table (140), the metadata mapping table (141), the element of value definition table (155) and the component of value definition table (156) required to initialize predictive model bots for each component of value at every level in the organization.

Bots are independent components of the application that have specific tasks to perform. In the case of predictive model bots, their primary task is determine the relationship between the item variables, item performance indicators and composite variables (collectively hereinafter, "the variables") and the components of value (and sub-components of value) by cluster at each level of the organization. A series of predictive model bots are initialized at this stage because it is impossible to know in advance which predictive model type will produce the "best" predictive model for the data from each commercial enterprise. The series for each model includes 9 predictive model bot types: neural network; CART; projection pursuit regression; generalized additive model (GAM), redundant regression network; boosted Naïve Bayes Regression; MARS; linear regression; and stepwise regression. The software in block 305 generates this series of predictive model bots for the levels of the organization shown in Table 22.

Table 22

| Predictive models by organization level |
|--|
| Organization: Enterprise variables relationship to organization revenue component of value by cluster Enterprise variables relationship to organization expense subcomponents of value by cluster Enterprise variables relationship to organization capital change subcomponents of value by cluster |
| Enterprise: Element variables relationship to enterprise revenue component of value by cluster Element variables relationship to enterprise expense subcomponents of value by cluster Element variables relationship to enterprise capital change subcomponents of value by cluster |
| Element of Value: Sub-element of value variables relationship to element of value |

Every predictive model bot contains the information shown in Table 23.

Table 23

| |
|---|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. Creation date (day, hour, minute, second) |
| 3. Mapping information |
| 4. Storage location |
| 5. Component or subcomponent of value |
| 6. Cluster (ID) |
| 7. Enterprise, Element or Sub-Element ID |
| 8. Predictive Model Type |
| 9. Variable 1 |
| ... |
| 9+n. Variable n |

After predictive model bots for each level in the organization are initialized, the bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). Once activated, the bots retrieve the required data from the appropriate table in the application database (50) and randomly partition the item variables, item performance indicators and composite variables into a training sets and a test set. The software in block 305 uses "bootstrapping" where the different training data sets are created by re-sampling with replacement from the original training set, so data records may occur more than once. The same sets of data will be used to train and then test each predictive model bot. When the predictive model bots complete their training and testing, processing advances to a block 306.

The software in block 306 uses a variable selection algorithm such as stepwise regression (other algorithms can be used) to combine the results from the predictive model bot analyses for each model to determine the best set of variables for each model. The models having the smallest amount of error as measured by applying the mean squared error algorithm to the test data are given preference in determining the best set of variables. As a result of this processing the best set of variables contain the item variables, item performance indicators and composite variables that correlate most strongly with changes in the components of value. The best set of variables will hereinafter be referred to as the "value drivers". Eliminating low correlation factors from the initial configuration of the vector creation algorithms increases the efficiency of the next stage of system processing. Other error algorithms alone or in combination may be substituted for the mean squared error algorithm. After the best set of variables have been selected and stored in the element variables table (158) for all models at all levels, the software in block 306 tests the independence of the value drivers at the enterprise, element and sub-element level before processing advances to a block 307.

The software in block 307 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation, a structure change or if the interaction between value drivers has changed from being highly correlated to being independent. If the calculation is not a new calculation, a structure change or a change to independent value driver status, then processing advances to a software block 310. Alternatively, if the calculation is new, a structure change or a

change to independent status, then processing advances to a software block 308. The software in block 308 checks the bot date table (149) and deactivates any induction bots with creation dates before the current system date. The software in block 308 then retrieves the information from the system settings table (140), the metadata mapping table (141), the component of value definition table (156) and the element variables table (158) as required to initialize induction model bots for each enterprise, element of value and sub-element of value at every level in the organization in accordance with the frequency specified by the user (20) in the system settings table (140) before processing advances to a block 309.

Bots are independent components of the application that have specific tasks to perform. In the case of induction bots, their primary tasks are to refine the item variable, item performance indicator and composite variable selection to reflect only causal variables and to produce formulas, (hereinafter, vectors) that summarize the relationship between the item variables, item performance indicators and composite variables and changes in the component or sub-component of value being examined. (Note: these variables are simply grouped together to represent an element vector when they are dependent). A series of induction bots are initialized at this stage because it is impossible to know in advance which induction algorithm will produce the "best" vector for the best fit variables from each model. The series for each model includes 4 induction bot types: entropy minimization, LaGrange, Bayesian and path analysis. The software in block 308 generates this series of induction bots for each set of variables stored in the element variables table (158) in the previous stage in processing. Every induction bot contains the information shown in Table 24.

203070 05404660

Table 24

| |
|---|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. Creation date (day, hour, minute, second) |
| 3. Mapping information |
| 4. Storage location |
| 5. Component or subcomponent of value |
| 6. Cluster (ID) |
| 7. Enterprise, Element or Sub-Element ID |
| 8. Variable Set |
| 9. Induction algorithm type |

After the induction bots are initialized by the software in block 308 processing passes to a software block 309. In block 309 bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). Once activated, they retrieve the element variable information for each model from the element variable table (158) and sub-divides the variables into two sets, one for training and one for testing. The same set of training data is used by each of the different types of bots for each model. After the induction bots complete their processing for each model, the software in block 309 uses a model selection algorithm to identify the vector that best fits the data for each enterprise, element or sub-element being analyzed. For the system of the present invention, a cross validation algorithm is used for model selection. The software in block 309 saves the the best fit vector in the vector table (159) in the application database (50) and processing returns to advances to a block 310. The software in block 310 tests the value drivers or vectors to see if there are "missing" value drivers that are influencing the results. If the software in block 310 doesn't detect any missing value drivers, then system processing advances to a block 322. Alternatively, if missing value drivers are detected by the software in block 310, then processing advances to a software block 321.

The software in block 321 prompts the user (20) via the variable identification window (710) to adjust the specification(s) for the affected enterprise, element of value or subelement of value. After the input from the user (20) is saved in the system settings

table (140) and/or the element of value definition table (155), system processing advances to a software block 323. The software in block 323 checks the in the system settings table (140) and/or the element of value definition table (155) to see if there any changes in structure. If there have been changes in the structure, then processing advances to a block 205 and the system processing described previously is repeated. Alternatively, if there are no changes in structure, then processing advances to a block 325.

The software in block 325 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change, then processing advances to a software block 329. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 326.

The software in block 326 checks the bot date table (149) and deactivates any option bots with creation dates before the current system date. The software in block 326 then retrieves the information from the system settings table (140), the metadata mapping table (141), the basic financial system database (143), the external database table (146) and the advanced finance system table (147) as required to initialize option bots for the organization, the industry and each enterprise in the organization before processing advances to a block 327.

Bots are independent components of the application that have specific tasks to perform. In the case of option bots, their primary tasks are to calculate the cost of capital (if the user (20) hasn't specified the cost of capital in the system settings table (140)) and value the real options for the industry, the organization, and each enterprise in the organization. The base cost of capital is calculated using a well known formula for the industry and each enterprise. The bots then use the data regarding the similarity of the "soft" asset profiles between the proposed real option activity and the existing industry, organization and enterprise profiles to determine the multiple on the cost of capital that will be used in valuing the real option. The closer the real option profile is to the existing profile, the closer the multiple is to one. If sufficient data is available, pattern matching algorithms can be used to replace the assessment by the user (20). After the cost of capital multiple has been determined, the value of the real option is calculated using

dynamic programming algorithms in a manner that is well known and stored in the real option value table (162). Real option values are calculated using dynamic programming algorithms. The real option can be valued using other algorithms including binomial, neural network or Black Scholes algorithms. The software in block 326 generates option bots for the industry, the organization and each enterprise in the organization.

Option bots contain the information shown in Table 25.

Table 25

| |
|---|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. Creation date (day, hour, minute, second) |
| 3. Mapping information |
| 4. Storage location |
| 5. Organization or Enterprise ID |
| 6. Real Option Type (Industry, Organization or Enterprise) |
| 7. Real Option |
| 8. Allocation % (if applicable) |

After the option bots are initialized by the software in block 326 processing passes to a block 327. In block 327 the bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated, the bots retrieve information for the organization, the industry and each enterprise in the organization from the basic financial system database (143), the external database table (146) and the advanced finance system table (147) as required to complete the option valuation. After the cost of capital multiple has been determined the value of the real option is calculated using dynamic programming algorithms in a manner that is well known. The resulting values are then saved in the real option value table (162) in the application database (50) before processing advances to a block 328.

The software in block 328 uses the item performance indicators produced by DEA analysis in blocks 304, 308 and 314 and the percentage of industry real options controlled by the enterprise to determine the allocation percentage for industry options. The more

dominant the organization and enterprise – as indicated by the industry rank for the intangible element indicators, the greater the allocation of industry real options. After the software in block 328 saves the information regarding the allocation of industry real options to the organization and each enterprise in the organization to the real option value table (162) in the application database (50) before advancing processing to a block 329.

The software in block 329 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change, then processing advances to a software block 333. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 330.

The software in block 330 checks the bot date table (149) and deactivates any cash flow bots with creation dates before the current system date. The software in block 326 then retrieves the information from the system settings table (140), the metadata mapping table (141) and the component of value definition table (156) as required to initialize cash flow bots for the organization and each enterprise in the organization in accordance with the frequency specified by the user (20) in the system settings table (140) before processing advances to a block 331.

Bots are independent components of the application that have specific tasks to perform. In the case of cash flow bots, their primary tasks are to calculate the cash flow for the organization and each enterprise in the organization for every time period where data is available and to forecast a steady state cash flow for the organization and each enterprise in the organization. Cash flow is calculated using a well known formula where cash flow equals period revenue minus period expense plus the period change in capital plus non-cash depreciation/amortization for the period. The steady state cash flow is calculated for the organization and each enterprise in the organization using forecasting methods identical to those disclosed previously in U.S. Patent 5,615,109 to forecast revenue, expenses, capital changes and depreciation separately before calculating the cash flow. The software in block 326 generates cash flow bots for the organization and each enterprise in the organization.

Every cash flow bot contains the information shown in Table 26.

Table 26

| |
|---|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. Creation date (day, hour, minute, second) |
| 3. Mapping information |
| 4. Storage location |
| 5. Organization or Enterprise ID |
| 6. Components of value |

After the cash flow bots are initialized by the software in block 330 processing passes to a block 331. In block 331 the bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated the bots retrieve the component of value information for the organization and each enterprise in the organization from the component of value definition table (156). The cash flow bots then complete the calculation and forecast of cash flow for the organization and each enterprise in the organization before saving the resulting values by period in the cash flow table (161) in the application database (50) before processing advances to a block 333.

The software in block 333 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change, then processing advances to a software block 343. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 341.

The software in block 341 checks the bot date table (149) and deactivates any element life bots with creation dates before the current system date. The software in block 341 then retrieves the information from the system settings table (140), the metadata mapping table (141) and the element of value definition table (155) as required to initialize element life bots for each element and sub-element of value in the organization before processing advances to a block 342.

Bots are independent components of the application that have specific tasks to perform. In the case of element life bots, their primary task is to determine the expected

life of each element and sub-element of value for each enterprise in the organization. There are three methods for evaluating the expected life of the elements and sub-elements of value. Elements of value that are defined by a population of members (such as: channel partners, customers, employees and vendors) will have their lives estimated by analyzing and forecasting the lives of the members of the population. The forecasting of member lives will be determined by the "best" fit solution from competing life estimation methods including the Iowa type survivor curves, Weibull distribution survivor curves, Gompertz-Makeham survivor curves, polynomial equations and the forecasting methodology disclosed in U.S. Patent 5,615,109. Elements of value (such as some parts of Intellectual Property – patents) that have legally defined lives will have their lives calculated using the time period between the current date and the expiration date of the element or sub-element. Finally, elements of value and sub-element of value (such as brand names, information technology and processes) that do not have defined lives and that do not consist of a collection of members will have their lives estimated by comparing the relative strength and stability of the element vectors with the relative stability of the enterprise CAP. The resulting values are stored in the element of value definition table (155) for each element and sub-element of value of each enterprise in the organization.

Every element life bot contains the information shown in Table 27.

Table 27

| |
|---|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. Creation date (day, hour, minute, second) |
| 3. Mapping information |
| 4. Storage location |
| 5. Element of Sub-Element of Value |
| 6. Life Estimation Method (population analysis, date calculation or relative CAP) |

After the element life bots are initialized by the software in block 341 processing passes to block 342. In block 342 the element life bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated, the bots retrieve information for each element and sub-element of value from the element of

value definition table (155) as required to complete the estimate of element life. The resulting values are then saved in the element of value definition table (155) in the application database (50) before processing advances to a block 343.

The software in block 343 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change, then processing advances to a software block 402. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 345.

The software in block 345 checks the bot date table (149) and deactivates any component capitalization bots with creation dates before the current system date. The software in block 341 then retrieves the information from the system settings table (140), the metadata mapping table (141) and the component of value definition table (156) as required to initialize component capitalization bots for the organization and each enterprise in the organization before processing advances to a block 346.

Bots are independent components of the application that have specific tasks to perform. In the case of component capitalization bots, their task is to determine the capitalized value of the components of value, forecast revenue, expense or capital requirements, for the organization and for each enterprise in the organization in accordance with the formula shown in Table 28.

Table 28

$$\text{Value} = F_{f1} / (1+K) + F_{f2} / (1+K)^2 + F_{f3} / (1+K)^3 + F_{f4} / (1+K)^4 + (F_{f4} \times (1+g)) / (1+K)^5 + (F_{f4} \times (1+g)^2) / (1+K)^6 \dots + (F_{f4} \times (1+g)^N) / (1+K)^{N+4}$$

Where:

| | |
|----------|--|
| F_{fx} | = Forecast revenue, expense or capital requirements for year x after valuation date (from advanced finance system) |
| N | = Number of years in CAP (from prior calculation) |
| K | = Cost of capital - % per year (from prior calculation) |
| g | = Forecast growth rate during CAP - % per year (from advanced finance system) |

After the capitalized value of every component and sub-component of value is complete, the results are stored in the component of value definition table (156) in the application database (50).

Every component capitalization bot contains the information shown in Table 29.

Table 29

| |
|---|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. Creation date (day, hour, minute, second) |
| 3. Mapping information |
| 4. Storage location |
| 5. Organization or Enterprise ID |
| 6. Component of Value (Revenue, Expense or Capital Change) |
| 7. Sub Component of Value |

After the component capitalization bots are initialized by the software in block 345 processing passes to block 346. In block 346 the component capitalization bots activate in accordance with the frequency specified by the user (20) in the system settings table

(140). After being activated, the bots retrieve information for each component and sub-component of value from the advanced finance system table (147) and the component of value definition table (156) as required to calculate the capitalized value of each component. The resulting values are then saved in the component of value definition table (156) in the application database (50) before processing advances to a block 347.

The software in block 347 checks the bot date table (149) and deactivates any valuation bots with creation dates before the current system date. The software in block 347 then retrieves the information from the system settings table (140), the metadata mapping table (141), the element of value definition table (155), the component of value definition table (156) as required to initialize valuation bots for each enterprise, element and sub-element of value in the organization before processing advances to a block 348.

Bots are independent components of the application that have specific tasks to perform. In the case of valuation bots, their task is to calculate the contribution of every enterprise, element of value and sub-element of value in the organization using the overall procedure outlined in Table 7. The first step in completing the calculation in accordance with the procedure outlined in Table 7, is determining the relative contribution of each enterprise and element of value by using a series of predictive models to find the best fit relationship between:

1. the enterprise contribution vectors and the organization components of value;
2. the element of value vectors and the enterprise components of value; and
3. the sub-element of value vectors and the element of value they correspond to.

The system of the present invention uses 9 different types of predictive models to determine relative contribution: neural network; CART; projection pursuit regression; generalized additive model (GAM), redundant regression network; boosted Naïve Bayes Regression; MARS; linear regression; and stepwise regression to determine relative contribution. The model having the smallest amount of error as measured by applying the mean squared error algorithm to the test data is the best fit model. The "relative contribution algorithm" used for completing the analysis varies with the model that was selected as the "best-fit". For example, if the "best-fit" model is a neural net model, then

20250709 09:40:50.010660

the portion of revenue attributable to each input vector is determined by the formula shown in Table 30.

Table 30

| | | | | |
|---|-------|-------|-------|-------|
| $k=m$ | $j=n$ | $j=n$ | $k=m$ | $j=n$ |
| $\left(\sum_{k=1}^m \sum_{j=1}^n I_{jk} \times O_k / \sum_{j=1}^n I_{jk} \right) / \sum_{k=1}^m \sum_{j=1}^n I_{jk} \times O_k$ | | | | |
| <p>Where</p> <p>I_{jk} = Absolute value of the input weight from input node j to hidden node k</p> <p>O_k = Absolute value of output weight from hidden node k</p> <p>m = number of hidden nodes</p> <p>n = number of input nodes</p> | | | | |

After the relative contribution of each enterprise, element of value and sub-element of value is determined, the results of this analysis are combined with the previously calculated information regarding element life and capitalized component value to complete the valuation of each: enterprise contribution, element of value and sub-element using the approach shown in Table 31.

Table 31

| Gross Value | Percentage | Element Life/CAP | Net Value |
|-----------------------------|------------|------------------|-------------------|
| Revenue value = \$120M | 20% | 80% | Value = \$19.2 M |
| Expense value = (\$80M) | 10% | 100% | Value = (\$8.0) M |
| Capital value = (\$5M) | 5% | 80% | Value = (\$0.2) M |
| Total value = \$35M | | | |
| Net value for this element: | | | Value = \$11.0 M |

The resulting values are stored in the element of value definition table (155) for each element and sub-element of value of each enterprise in the organization.

Every valuation bot contains the information shown in Table 32.

Table 32

| |
|---|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. Creation date (day, hour, minute, second) |
| 3. Mapping information |
| 4. Storage location |
| 5. Enterprise Contribution, Element of Value or Sub-Element of Value |
| 6. Organization, Enterprise or Element of Value ID |

After the valuation bots are initialized by the software in block 347 processing passes to block 348. In block 348 the valuation bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated, the bots retrieve information from the element of value definition table (155) and the component of value definition table (156) as required to complete the valuation. The resulting values are then saved in the element of value definition table (155) in the application database (50) before processing advances to a block 349.

The software in block 349 checks the bot date table (149) and deactivates any residual bots with creation dates before the current system date. The software in block 349 then retrieves the information from the system settings table (140), the metadata mapping table (141) and the element of value definition table (155) as required to initialize residual bots for each enterprise in the organization.

Bots are independent components of the application that have specific tasks to perform. In the case of residual bots, their task is to retrieve data from the as required from the element of value definition table (155) and the component of value definition table (156) and then calculate the residual going concern value for the organization and each enterprise in the organization in accordance with the formula shown in Table 33.

Table 33

| |
|---|
| $\text{Residual Going Concern Value} = \text{Total Current-Operation Value} - \sum \text{Financial Asset Values} - \sum \text{Elements of value}$ |
|---|

Every residual bot contains the information shown in Table 34.

Table 34

| |
|---|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. Creation date (day, hour, minute, second) |
| 3. Mapping information |
| 4. Storage location |
| 5. Organization or Enterprise ID |

After the residual bots are initialized by the software in block 348 processing passes to block 349. In block 349 the residual bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated, the bots retrieve information from the element of value definition table (155) and the component of value definition table (156) as required to complete the residual calculation for the organization or enterprise. After the calculation is complete, the resulting values are then saved in the element of value definition table (155) in the application database (50) before processing advances to a block 402.

ANALYZE MARKET SENTIMENT

The flow diagram in FIG. 7 details the processing that is completed by the portion of the application software (400) that analyzes the market sentiment for the enterprises in the organization. Processing begins in a software block 402.

The software in block 402 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change, then processing

advances to a software block 409. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 404.

The software in block 404 checks the bot date table (149) and deactivates any sentiment calculation bots with creation dates before the current system date. The software in block 404 then retrieves the information from the system settings table (140), the metadata mapping table (141), the external database table (146), the element of value definition table (155), the component of value definition table (156) and the real option value table (162) as required to initialize sentiment calculation bots for each enterprise in the organization.

Bots are independent components of the application that have specific tasks to perform. In the case of sentiment calculation bots, their task is to retrieve data as required from: the external database table (146), the element of value definition table (155), the component of value definition table (156) and the real option value table (162) then calculate the sentiment for each enterprise in the organization in accordance with the formula shown in Table 35.

Table 35

| |
|--|
| $\text{Sentiment} = \text{Total Market Value} - \text{Total Current-Operation Value} - \sum \text{Real Option Values}$ |
|--|

Every sentiment calculation bot contains the information shown in Table 36.

Table 36

| |
|---|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. Creation date (day, hour, minute, second) |
| 3. Mapping information |
| 4. Storage location |
| 5. Enterprise ID |

After the sentiment calculation bots are initialized by the software in block 404 processing passes to block 405. In block 405 the sentiment calculation bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated, the bots retrieve information from the external database table (146), the element of value definition table (155), the component of value definition table (156) and the real option value table (162) as required to complete the sentiment calculation for each enterprise. After the calculation is complete, the resulting values are then saved in the enterprise sentiment table (166) in the application database (50) before processing advances to a block 409.

The software in block 409 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change, then processing advances to a software block 412. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 410.

The software in block 410 checks the bot date table (149) and deactivates any sentiment factor bots with creation dates before the current system date. The software in block 410 then retrieves the information from the system settings table (140), the metadata mapping table (141), the external database table (146), the element of value definition table (155), the component of value definition table (156) and the real option value table (162) as required to initialize sentiment factor bots for each enterprise in the organization.

Bots are independent components of the application that have specific tasks to perform. In the case of sentiment factor bots, their primary task is to calculate sentiment related attributes including cumulative total value, the period to period rate of change in value, the rolling average value, a series of time lagged values as well as pre-specified combinations of variables called composite variables. The bots also use attribute derivation algorithms such as the AQ program to create combinations of the variables that weren't pre-specified for combination. While the AQ program is used in the preferred embodiment of the present invention, other attribute derivation algorithms such as the LINUS algorithms, may be used to the same effect. The newly calculated sentiment

factors are stored in the sentiment factor table (169) before processing advances to a block 411.

Every sentiment factor bot contains the information shown in Table 37.

Table 37

| |
|---|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. Creation date (day, hour, minute, second) |
| 3. Mapping information |
| 4. Storage location |
| 5. Enterprise ID |

After the sentiment factor bots are initialized by the software in block 410 processing passes to block 411. In block 411 the sentiment factor bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated, the bots retrieve information from the external database table (146), the element of value definition table (155), the component of value definition table (156) and the real option value table (162) as required to generate the sentiment factors for each enterprise. After the calculation is complete, the resulting values are then saved in the sentiment factors table (169) in the application database (50) before processing advances to a block 412.

The software in block 412 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change, then processing advances to a software block 502. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 413.

The software in block 413 checks the bot date table (149) and deactivates any sentiment analysis bots with creation dates before the current system date. The software in block 413 then retrieves the information from the system settings table (140), the metadata mapping table (141), the external database table (146), the element of value definition table (155), the component of value definition table (156), the real option value

table (162), the enterprise sentiment table (166) and the sentiment factors table (169) as required to initialize sentiment analysis bots for each enterprise in the organization.

Bots are independent components of the application that have specific tasks to perform. In the case of sentiment analysis bots, their primary task is determine the relationship between sentiment factors and the calculated sentiment for each enterprise in the organization. A series of predictive model bots are initialized at this stage because it is impossible to know in advance which predictive model type will produce the "best" predictive model for the data from each commercial enterprise. The series for each model includes 9 predictive model bot types: neural network; CART; projection pursuit regression; generalized additive model (GAM), redundant regression network; boosted Naïve Bayes Regression; MARS; linear regression; and stepwise regression.

Every sentiment analysis bot contains the information shown in Table 38.

Table 38

| |
|---|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. Creation date (day, hour, minute, second) |
| 3. Mapping information |
| 4. Storage location |
| 5. Enterprise ID |

After the sentiment analysis bots are initialized by the software in block 413 processing passes to block 414. In block 411 the sentiment analysis bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated, the bots retrieve information from the the system settings table (140), the metadata mapping table (141), the enterprise sentiment table (166) and the sentiment factors table (169) and randomly partition sentiment factors for each enterprise into a training set and a test set. The software in block 414 uses "bootstrapping" where the different training data sets are created by re-sampling with replacement from the original training set, so data records may occur more than once. The same sets of data will be used to train and then test each predictive model bot. When the predictive model bots

complete their training and testing, the resulting sets of "best fit" factors are then saved in the sentiment factors table (169) in the application database (50) before processing advances to a block 415.

The software in block 415 combines the results from the sentiment analysis from each bot type to determine the best set of sentiment factors for each enterprise. The models having the smallest amount of error as measured by applying the mean squared error algorithm to the test data are given preference in determining the best set of variables. As a result of this processing the best set of variables contain the sentiment factors that correlate most strongly with changes in the components of value. The best set of variables will hereinafter be referred to as the "sentiment drivers". The software in block 415 saves an indicator in each item record identifying the sentiment factors that are "sentiment drivers" before processing advances to block 502.

DISPLAY AND PRINT RESULTS

The flow diagram in FIG. 8 details the processing that is completed by the portion of the application software (500) that creates and displays financial management reports, optionally prints financial management reports and optionally trades company equity securities. The financial management reports use the Value Map® report format to summarize information about the categories of business value for the organization and each enterprise in the organization. If there are prior valuations, then a Value Creation report will be created to highlight changes in the categories of business value during the period between the prior valuation and the current valuation date.

System processing in this portion of the application software (900) begins in a block 502. The software in block 502 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change, then processing advances to a software block 505. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 504.

The software in block 504 checks the bot date table (149) and deactivates any report bots with creation dates before the current system date. The software in block 504 then

retrieves the information from the system settings table (140) and the report table (164) as required to determine the format (Value Map® & Value Creation format and/or traditional: balance sheet, income & cash flow statement format) and type of report (text or graphical) bots that need to be created for the organization, each enterprise in the organization and the sub-elements of value before processing advances to block 505.

Bots are independent components of the application that have specific tasks to perform. In the case of report bots, their primary tasks are to: retrieve data from the system settings table (140), the basic finance system table (143), the advanced finance system table (147), the element of value definition table (155), the component of value definition table (156) and the real option value table (162), calculate market equity using the formula shown in Table 39 and generate the reports in the specified formats for the specified time period(s).

Table 39

| |
|--|
| $\begin{aligned} \text{Market Equity} &= (\text{Current Operation Value}) \\ &+ (\sum \text{Real Option Values}) - (\sum \text{Short Term Liabilities}) \\ &- (\sum \text{Contingent \& Long Term Liabilities}) - (\text{Book Value of Equity}) \end{aligned}$ |
|--|

*calculated in accordance with GAAP

Every report bot contains the information shown in Table 40.

Table 40

| |
|---|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. Creation date (day, hour, minute, second) |
| 3. Mapping information |
| 4. Storage location |
| 5. Organization, Enterprise or Element of ValueID |
| 6. Report Format (text or graphical) |
| 7. Report Type (Value Map®/Value Creation format or traditional format) |

The general format of the Value Map® Reports is shown in Table 41 and Table 42.

Table 41

| Value Map™ Report XYZ Corporation | | |
|--|-----------------------------|-----------------------------|
| <u>ASSETS</u> | <u>12/31/19XX</u> | <u>12/31/XXXX</u> |
| Current Operation: | | |
| Financial Assets | | |
| - Cash and Marketable Securities: | \$7,871,230 | \$15,097,057 |
| - Accounts Receivable | \$39,881,200 | \$42,234,410 |
| - Inventory | \$19,801,140 | \$21,566,540 |
| - Property, Plant & Equipment | \$22,800,000 | \$21,221,190 |
| - Prepaid Expenses | <u>\$2,071,440</u> | <u>\$1,795,890</u> |
| Subtotal Current Operation Assets: | \$92,425,010 | \$101,915,087 |
| Cash Generating "Soft" Assets | | |
| - Brandnames | \$17,000,000 | \$12,000,000 |
| - Customer Base | \$62,000,000 | \$49,500,000 |
| - Employees | \$10,750,000 | \$8,250,000 |
| - Strategic Alliances | \$33,250,000 | \$33,500,000 |
| - Vendors | \$11,500,000 | \$9,750,000 |
| - General Going Concern Value | <u>\$31,250,000</u> | <u>\$31,750,000</u> |
| Subtotal Cash Generating Assets | \$165,750,000 | \$144,750,000 |
| Subtotal Current Operation | \$258,175,010 | \$246,665,087 |
| Real Options: | | |
| GUI Market Option | \$12,500,000 | \$10,000,000 |
| IPX Market Option | <u>\$17,000,000</u> | <u>\$12,500,000</u> |
| Subtotal Enterprise Options | \$29,500,000 | \$22,500,000 |
| Industry Growth Options: | <u>\$80,000,000</u> | <u>\$60,000,000</u> |
| Subtotal Real Options | \$109,500,000 | \$82,500,000 |
| Total Assets & Options | \$367,675,010 | \$329,165,087 |
| Market Sentiment | \$27,123,116 | \$18,273,698 |
| Total Market Value | <u>\$394,798,126</u> | <u>\$347,438,785</u> |
| Copyright, Jeff S. Eder 1999, All Rights Reserved | | |

2008070" 05404650

Table 42

**Value Map™ Report
XYZ Corporation**

LIABILITIES & SHAREHOLDER EQUITY**Liabilities:**

| | | |
|---|-----------------------------|-----------------------------|
| - Accounts Payable | \$15,895,585 | \$18,879,949 |
| - Salaries Payable | \$8,766,995 | \$10,468,305 |
| - Short Term Debt, Notes Payable | \$20,189,900 | \$11,506,130 |
| - Taxes Payable | <u>\$12,430,120</u> | <u>\$9,099,880</u> |
| Subtotal Short Term Liabilities | \$57,282,600 | \$49,954,264 |
| Contingent Liabilities | \$5,100,000 | \$4,800,000 |
| Long Term Debt | \$17,800,000 | \$20,916,650 |
| Total Liabilities | \$80,182,600 | \$75,670,914 |
| Shareholder's Equity: | | |
| - Stock | \$2,000,000 | \$2,000,000 |
| - Market Equity | \$27,123,116 | \$18,273,698 |
| - Retained Earnings | \$15,342,410 | \$29,044,173 |
| - Future Earnings | <u>\$270,150,000</u> | <u>\$222,450,000</u> |
| Total Shareholder's Equity | \$314,615,526 | \$271,767,871 |
| Total Liabilities & Shareholder Equity | <u><u>\$394,798,126</u></u> | <u><u>\$347,438,785</u></u> |

Copyright, Jeff S. Eder 1999, All Rights Reserved

After the report bots are initialized by the software in block 504 processing passes to a block 505. In block 505 the bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated, the bots retrieve information for the organization, enterprise or element of value from the element of value definition table (155), the component of value definition table (156) and the real option

value table (1) as required to complete the report in accordance with the pre-specified format. The resulting reports are then saved in the report table (164) in the application database (50). The software in block 505 creates and displays all Value Map® reports and Value Creation Statement reports the user (20) requests using the report selection and display data window (705) in the general format shown in Table 41. Graphical reports such as those in a Hyperbolic Tree format that have been saved over time can be displayed like a "movie" shows the evolution of value over time. The software in block 505 also prompts the user (20) using the report selection and display data window (705) to select reports for printing. After the user's input regarding reports to print has been stored in the reports table (164), processing advances to block 507. If the user doesn't provide any input, then only the default reports specified by the user (20) in the system settings table (140) will be produced for storage.

The software in block 507 checks the reports tables (164) to determine if any reports have been designated for printing. If reports have been designated for printing, then processing advances to a block 506. The software in block 506 sends the designated reports to the printer (118). After the reports have been sent to the printer (118), processing advances to a software block 509. Alternatively, if no reports were designated for printing then processing advances directly from block 507 to block 509.

The software in block 509 checks the system settings table (140) in the application database (50) to determine if trading in enterprise equity is authorized. If trading in enterprise equity is not authorized, then processing advances to a software block 507. Alternatively, if trading in enterprise equity is authorized, then processing advances to a software block 510.

The software in block 510 retrieves information from the system settings table (140) and the advanced finance system table (147) that is required to calculate the minimum amount of cash that will be available for investment in enterprise equity during the next 12 month period. The system settings table (140) contains the minimum amount of cash and available securities that the user (20) indicated was required for enterprise operation while the advanced finance system table (147) contains a forecast of the cash balance for the enterprise for each period during the next 12 months. After the amount of available cash

for each enterprise is calculated and stored in the equity purchase table (165), processing advances to a software block 511.

The software in block 511 checks the equity purchase table (165) and enterprise sentiment table (166) to see if there is negative sentiment in any enterprise with available cash. If there are no enterprises with negative sentiment and available cash, then processing advances a software block 602. Alternatively, if there are enterprises with available cash and negative sentiment, then processing advances to a software block 512.

The software in block 512, retrieves the current enterprise equity price from the external database table (146), calculates the number of shares that can be purchased using the available cash and then generates a purchase order for the number of shares that can be purchased. The software in block 512 then prompts the user (20) via the purchase shares and confirm data window (706) to confirm the purchase. Once the user (20) confirms the equity purchase, the software in block 512 retrieves the on-line equity account information from the system settings table (140) and transmits and confirms the order to purchase the shares with the on-line broker via the network (45). The details of equity purchase transaction and confirmation are saved in the equity purchase table (156) before processing advances to block 602.

GENERATE AND ANALYZE VALUE IMPROVEMENTS

The flow diagram in FIG. 9 details the processing that is completed by the portion of the application software (600) that generates and analyzes value improvements. Processing in this portion of the application starts in software block 602.

The software in block 602 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change, then processing advances to a software block 606. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 603.

The software in block 603 checks the bot date table (149) and deactivates any improvement bots with creation dates before the current system date. The software in block 603 then retrieves the information from the system settings table (140), the soft asset system table (148), the element of value definition table (155) and the component of value definition table (156) as required to initialize improvement bots before processing advances to a block 604.

Bots are independent components of the application that have specific tasks to perform. In the case of improvement bots, their primary task is to analyze and prioritize potential changes to value drivers for each enterprise in the organization. The analysis of value driver changes closely mirrors the calculation of profit improvement that was completed in the related US Patent 5,615,109 a "Method of and System for Generating Feasible, Profit Maximizing Requisition Sets". The capital efficiency of the potential improvements identified by the improvement bots is evaluated in accordance with the formula shown in Table 43.

Table 43

| | Capital Change (+) | Capital Change (-) |
|--------------------|--|--|
| Capital efficiency | $\frac{\text{Revenue}\Delta - \text{Expense}\Delta}{\text{Capital } \Delta}$ | $\frac{\text{Revenue}\Delta - \text{Expense}\Delta}{\text{Capital } \Delta}$ |
| Where: | Revenue Δ = revenue impact of 1% change in value driver Expense Δ = expense impact of 1% change in value driver Capital Δ = capital impact of 1% change in value driver | |

The software in block 604 generates a list of potential improvements for each element of value defined and measured by the system of the present invention.

Every improvement bot contains the information shown in Table 44.

Table 44

| |
|---|
| 1. Unique ID number (based on date, hour, minute, second of creation) |
| 2. Creation date (day, hour, minute, second) |
| 3. Mapping information |
| 4. Storage location |
| 5. Element of ValueID |
| 6. Soft Asset System |
| 7. Value Driver |

After the improvement bots are initialized by the software in block 603 processing passes to a block 604. In block 604 the bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated, the bots retrieve information for the element of value from the system settings table (140), the soft asset system table (148), the element of value definition table (155) and the component of value definition table (156) as required to complete the analyses in accordance with the formula shown in Table 40. The soft asset management system that corresponds to the element of value being analyzed may also have generated a list of potential improvements. If it has generated a list, these improvements are analyzed in the same manner that the improvements generated by the system of the present invention are analyzed. The resulting list of prioritized improvements are then saved in the value driver change table (167) in the application database (50) before processing advances to a block 605.

The software in block 605 prepares a list of the potential value improvements in capital efficiency order and prompts the user (20) via a value driver and structure change window (707) to modify and/or select the improvements and/or structure changes that should be included in the revised forecast. If the user (20) chooses not to enter any selections, then the software in block 605 will select the potential improvements that produce the most benefit within the constraints imposed by the available cash. The information regarding the improvement selections made by the user (20) or the system are stored in the value driver change table (167) in the application database (50). In a similar fashion, if the user made any changes to the structure, the information regarding

the new change is stored in the system settings table (140) before processing advances to a software block 606.

The software in block 606 checks the system settings table (140) in the application database (50) to determine if the current calculation is a structure change. If the calculation is new or a structure change, then processing advances to software block 204 and the processing described above is repeated. Alternatively, if the calculation is not a structure change, then processing advances to a software block 610.

The software in block 610 retrieves information from the system settings table (140), the element of value definition table (155), the component of value definition table (156) and the value driver change table (167) as required to define and initialize a probabilistic simulation model. The preferred embodiment of the probabilistic simulation model is a Markov Chain Monte Carlo model, however, other simulation models can be used with similar results. The information defining the model is then stored in the simulation table (168) before the software in block 610 iterates the model as required to ensure the convergence of the frequency distribution of the output variables. After the simulation calculations have been completed, the software in block 610 saves the resulting information in the simulation table (168) before displaying the results of the simulation to the user (20) via a Value Mentor™ Reports data window (708) that uses a summary Value Map™ report format to display the mid point and the range of estimated future values for the various elements of each enterprise and the changes in value drivers, user-specified or system generated, that drove the future value estimate. The user (20) is prompted to indicate when the examination of the displayed report is complete and to indicate if any reports should be printed. If the user (20) doesn't provide any information regarding reports to display or print, then no reports are displayed or printed at this point and system processing continues. The information entered by the user (20) is entered in to the report table (164) before processing advances to a block 611.

The software in block 611 checks the reports tables (164) to determine if any additional reports have been designated for printing. If additional reports have been designated for printing, then processing advances to a block 612 which prepares and sends the designated reports to the printer (118). After the reports have been sent to the

printer (118), processing advances to a software block 614. If the software in block 611 determines that no additional reports have been designated for printing, then processing advances directly to block 614.

The software in block 614 checks the system settings table (140) in the application database (50) to determine if the current calculation is a continuous calculation. If the calculation is a continuous calculation, then processing advances to software block 204 where the processing described previously is repeated continuously. Alternatively, if the calculation is not continuous, then processing advances to a software block 615 where processing stops.

Thus, the reader will see that the system and method described above transforms extracted transaction data, corporate information and information from the internet into detailed valuations for an organization, the enterprises in the organization and for specific elements of value within the enterprise. The level of detail contained in the business valuations allows users of the system to monitor and manage efforts to improve the value of the business in a manner that is superior to that available to users of traditional accounting systems and business valuation reports.

While the above description contains many specificity's, these should not be construed as limitations on the scope of the invention, but rather as an exemplification of one preferred embodiment thereof. Accordingly, the scope of the invention should be determined not by the embodiment illustrated, but by the appended claims and their legal equivalents.

2025-05-04 09:40:40